Submitted, ACM Transactions on Embedded Computing Systems
http://www.cds.caltech.edu/~murray/papers/wmml09-tecs.html

# Verification of Periodically Controlled Hybrid Systems: Application to An Autonomous Vehicle

TICHAKORN WONGPIROMSARN
California Institute of Technology
SAYAN MITRA
University of Illinois at Urbana Champaign
and
RICHARD M. MURRAY and ANDREW LAMPERSKI
California Institute of Technology

This paper introduces Periodically Controlled Hybrid Automata (PCHA) for modular specification of hybrid control systems. In a PCHA, *control* actions that change the control input to the plant occur roughly periodically, while other actions that update the state of the controller may occur in the interim, changing the set-point of the system. Such actions could model, for example, sensor updates and information received from higher-level planning modules that change the set-point of the controller. Based on periodicity and subtangential conditions, a new sufficient condition for verifying invariant properties of PCHAs is presented. Checking these conditions can be automated using, for example, the constraint-based approach, quantifier elimination, or sum of squares decomposition. The proposed technique is used to verify safety and progress properties of the planner-controller subsystem of an autonomous ground vehicle. Geometric properties of planner generated paths are derived which guarantee that such paths can be safely followed by the controller.

Categories and Subject Descriptors: D.2.4 [**Software Engineering**]: Software/Program Verification—*Correctness proofs*; *Formal methods*; F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs—*Invariants*; *Specification techniques*; I.2.9 [**Artificial Intelligence**]: Robotics—*Autonomous vehicles*

General Terms: Verification

Additional Key Words and Phrases: Hybrid systems

## 1. INTRODUCTION

Design bugs in embedded systems can be fairly subtle and may arise from the unforeseen interactions among the computing, communication, and control subsystems. Consider, for example, the embedded computing system of the autonomous vehicle *Alice* built at Caltech. Alice successfully accomplished two of the three tasks

Author's address: T. Wongpiromsarn, California Institute of Technology, Mail Code 104-44, 1200 E. California Blvd., Pasadena, CA 91125.
This work is partially supported by AFOSR through the MURI program.

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **2009** | | **00-00-2009 to 00-00-2009** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **Verification of Periodically Controlled Hybrid Systems: Application to An Autonomous Vehicle** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **California Institute of Technology,Thomas J. Watson Laboratory of Applied Physics,Pasadena,CA,91125** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES
**ACM Transactions on Embedded Computing Systems, 2009 (Submitted). U.S. Government or Federal Rights License**

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT **unclassified** | b. ABSTRACT **unclassified** | c. THIS PAGE **unclassified** | **Same as Report (SAR)** | **39** | |

at the National Qualifying Event of the 2007 DARPA Urban Challenge [Burdick et al. 2007], [Wongpiromsarn and Murray 2008], [DuToit et al. 2008]. In executing the third task, which involved making left-turns while merging into traffic, its behavior was unsafe and almost led to a collision. Alice was stuck at the corner of a sharp turn dangerously stuttering in the middle of an intersection. It was later diagnosed that this behavior was caused by bad interactions between the *reactive obstacle avoidance subsystem (ROA)* and the relatively slowly reacting *path planner*. The planner incrementally generates a sequence of waypoints based on the road map, obstacles, and the mission goals. The ROA is designed to rapidly decelerate the vehicle when it gets too close to (possibly dynamic) obstacles or when the deviation from the planned path gets too large. Finally, to protect the vehicle steering system, Alice's low-level controller limits the rate of steering at low speeds. Thus, accelerating from a low speed, if the planner produces a path with a sharp left turn, the controller is unable to execute the turn closely. Alice deviates from the path; the ROA activates and slows it down. This cycle continues leading to stuttering.

The above example illustrates how the design of reliable embedded systems inherit the difficulties involved in designing both control systems and distributed (concurrent) computing systems. The described design bug manifests as the undesirable behavior only under a very specific set of conditions and only when the controller, the ROA, and the vehicle interact in a very specific manner. Therefore, such a bug had never got discovered by thousands of hours of our extensive simulations and over three hundred miles of field testing. Formal methods provide tools and techniques for uncovering such subtle design bugs and mathematically prove correctness of designs. More recently, formal techniques have also been used to automatically generate controllers that are correct by construction [Kloetzer and Belta 2006], [Fainekos et al. 2009].

The hybrid system formalism [Alur et al. 1995], [Kaynar et al. 2005] provides a rich mathematical language for specifying embedded systems where computing and control components interact with physical processes. The algorithmic verification problem for hybrid systems with general dynamics is known to be computationally hard [Henzinger et al. 1995]. Restricted subclasses that are amenable to algorithmic analysis have been identified, such as the rectangular-initialized hybrid automata [Henzinger et al. 1995], o-minimal hybrid automata [Lafferriere et al. 1999], and more recently planar [Prabhakar et al. 2008] and stormed [Vladimerou et al. 2008] hybrid automata. Although these restricted subclasses improve our understanding of the decidability frontier for hybrid systems, the imposed restrictions are artificial. That is, they are not representative of structures that arise in real engineered systems. For example, initialized hybrid automata require the continuous state of the system to be reset every time the automaton enters a new mode (control state). STORMED hybrid automata, on the other hand, require all the vector fields and reset maps to be monotonic with respect to a certain fixed direction.

While real world hybrid systems are large and complex, they are also engineered, and hence, have more structure than general hybrid automata [Alur et al. 1995]. With the motivation of abstractly capturing a common design pattern in embedded

control systems, such as Alice, and other motion control systems [Mitra et al. 2003], in this paper we study a new subclass of hybrid automata.

Two main contributions of this paper are the following[1]: First, we define a class of hybrid control systems in which certain *control actions* occur roughly periodically. Each control action sets the *controlling output* that drives the plant or the physical process. In the interval between two control actions the state of the plant evolves continuously with the control input set by the first. Also, in the same interval, other discrete actions may occur updating the state of the system. For example, such discrete changes may correspond to sensor inputs and changes of the waypoint or the set-point of the controller. These changes may in turn influence the computation of the next control action.

For this class of *periodically controlled hybrid systems*, we present a sufficient condition for verifying invariant properties. The key requirement in applying this condition is to identify a collection of subset(s) $C$ of the candidate invariant set $\mathcal{I}$, such that if the control action occurs when the system state is in $C$, then the subsequent control output guarantees that the system remains in $\mathcal{I}$ for the next period. The technique does not require solving the differential equations; instead, it relies on checking conditions on the periodicity and the subtangential condition at the boundary of $\mathcal{I}$. We show how these checks can be automated using sum of squares decomposition and semidefinite programming [Prajna et al. 2002]. These formulations are illustrated by analyzing an example in which an invariant is automatically determined using the constraint-based approach presented in [Gulwani and Tiwari 2008]. We believe that other techniques for finding invariants, for example those presented in [Platzer and Clarke 2008], [Sankaranarayanan et al. 2008], could also be effectively used for computing invariants of PCHAs. The findings from this direction of research will be reported in a future paper.

Secondly, we apply the above technique to verify the safety and progress properties of the planner-controller subsystem of Alice. First, we verify a family of invariants $\{\mathcal{I}_k\}_{k\in\mathbb{N}}$ using the above-mentioned technique. Then, we determine a sequence of shrinking $\mathcal{I}_k$'s as the vehicle makes progress along the planned path. From these shrinking invariants, we are able to deduce safety. That is, the deviation—distance of the vehicle from the planned path—remains within a certain constant bound. In the process, we also derive geometric properties of planner paths that guarantee that they can be followed safely by the vehicle.

The remainder of the paper is organized as follows: In Section 2, we briefly present the key definitions for the hybrid I/O automaton framework. In Section 3, we present PCHA and a sufficient condition for proving invariance. In this section, we also present the formulation of the sufficient conditions as a sum of squares optimization problem for automatic verification. In Sections 4 and 5, we present the formal model and verification of Alice's Controller-Vehicle subsystem.

## 2. PRELIMINARIES

We use the Hybrid Input/Output Automata (HIOA) framework of [Lynch et al. 2003; Kaynar et al. 2005] for modelling hybrid systems and the state model-based

[1]The preliminary results of this paper were published in [Wongpiromsarn et al. 2009].

notations introduced in [Mitra 2007]. A Structured Hybrid I/O Automaton (SHIOA) is a non-deterministic state machine whose state may change instantaneously through a transition, or continuously over an interval of time following a *trajectory*.

A variable structure is used for specifying the states of an SHIOA. Let $V$ be a set of variables. Each variable $v \in V$ is associated with a *type* which defines the set of values $v$ can take. The set of valuations of $V$ is denoted by $\text{val}(V)$. For a valuation $\mathbf{v} \in \text{val}(V)$ of set of variables $V$, its restriction to a subset of variables $Z \subseteq V$ is denoted by $\mathbf{v} \lceil Z$. A variable may be *discrete* or *continuous*[2]. Typically, discrete variables model protocol or software state, and continuous variables model physical quantities such as time, position, and velocity.

A *trajectory* for a set of variables $V$ models continuous evolution of the values of the variables over an interval of time. Formally, a trajectory $\tau$ is a map from a left-closed interval of $\mathbb{R}_{\geq 0}$ with left endpoint 0 to $\text{val}(V)$. The domain of $\tau$ is denoted by $\tau.dom$. The *first state* of $\tau$, $\tau.\mathsf{fstate}$, is $\tau(0)$. A trajectory $\tau$ is *closed* if $\tau.dom = [0, t]$ for some $t \in \mathbb{R}_{\geq 0}$, in which case we define the *last time* of $\tau$, $\tau.\mathsf{ltime} \triangleq t$, and the *last state* of $\tau$, $\tau.\mathsf{lstate} \triangleq \tau(t)$. For a trajectory $\tau$ for $V$, its restriction to a subset of variables $Z \subseteq V$ is denoted by $\tau \downarrow Z$.

For given sets of input $U$, output $Y$, and internal $X$ variables, a *state model* $\mathcal{S}$ is a triple $(\mathscr{F}, Inv, Stop)$, where (a) $\mathscr{F}$ is a collection of Differential and Algebraic Inequalities (DAIs) involving the continuous variables in $U, Y$, and $X$, and (b) $Inv$ and $Stop$ are predicates on $X$ called *invariant condition* and *stopping condition* of $\mathcal{S}$. Components of $\mathcal{S}$ are denoted by $\mathscr{F}_{\mathcal{S}}$, $Inv_{\mathcal{S}}$ and $Stop_{\mathcal{S}}$. $\mathcal{S}$ defines a set of trajectories, denoted by $traj(\mathcal{S})$, for the set of variables $V = X \cup U \cup Y$. A trajectory $\tau$ for $V$ is in the set $trajs(\mathcal{S})$ iff

(a) the discrete variables in $X \cup Y$ remain constant over $\tau$;

(b) the restriction of $\tau$ on the continuous variables in $X \cup Y$ satisfies all the DAIs in $\mathscr{F}_{\mathcal{S}}$;

(c) at every point in time $t \in dom(\tau)$, $(\tau \downarrow X)(t) \in Inv$; and

(d) if $(\tau \downarrow X)(t) \in Stop$ for some $t \in dom(\tau)$, then $\tau$ is closed and $t = \tau.\mathsf{ltime}$.

A *Structured Hybrid I/O Automaton* is a state machine that uses a collection of state models for specifying its trajectories.

***Definition* 2.1.** A *Structured Hybrid I/O Automaton (SHIOA)* $\mathcal{A}$ is a tuple $(V, Q, Q_0, A, \mathcal{D}, \mathscr{S})$ where

(a) $V$ is a set of variables partitioned into sets of *internal* or *state* variables $X$, *output* variables $Y$ and *input* variables $U$;

(b) $Q \subseteq \text{val}(X)$ is a set of *states* and $Q_0 \subseteq Q$ is a nonempty set of *start states*;

(c) $A$ is a set of actions partitioned into sets of *internal* $H$, *output* $O$ and *input* $I$ actions;

(d) $\mathcal{D} \subseteq Q \times A \times Q$ is a set of *discrete transitions*; and

(e) $\mathscr{S}$ is a collection of *state models* for $U$, $Y$, and $X$, such that for every $\mathcal{S}, \mathcal{S}' \in \mathscr{S}$, $Inv_{\mathcal{S}} \cap Inv_{\mathcal{S}'} = \emptyset$ and $Q \subseteq \bigcup_{\mathcal{S} \in \mathscr{S}} Inv_{\mathcal{S}}$.

[2]See [Mitra 2007] for formal definition of these variable dynamic types.

In addition, $\mathcal{A}$ satisfies the following axioms:

**E1** Every input action is enabled at every state.

**E2** Given any trajectory $v$ of the input variables $U$, any $\mathcal{S} \in \mathscr{S}$, and $\mathbf{x} \in Inv_{\mathcal{S}}$, there exists $\tau \in trajs(\mathcal{S})$ starting from $\mathbf{x}$, such that either (a) $\tau \downarrow U = v$, or (b) $\tau \downarrow U$ is a proper prefix of $v$ and some action in $H \cup O$ is enabled at $\tau.\mathsf{lstate}$.

**E1** is the standard action nonblocking axiom of I/O automata. **E2** is a nonblocking axiom for individual state models: given any trajectory $v$ of the input variables and any state model, either time can elapse for the entire duration of $v$, or time elapses to a point at which some local action of $\mathcal{A}$ is enabled.

For a set of state variables $X$, a state $\mathbf{x}$ is an element of $\mathrm{val}(X)$. We denote the valuation of a variable $y \in X$ at state $\mathbf{x}$, by the usual (.) notation $\mathbf{x}.y$. A transition $(\mathbf{x}, a, \mathbf{x}') \in \mathcal{D}$ is written in short as $\mathbf{x} \rightarrow^{a}_{\mathcal{A}} \mathbf{x}'$ or as $\mathbf{x} \xrightarrow{a} \mathbf{x}'$ when $\mathcal{A}$ is clear from the context. An action $a$ is said to *enabled* at $\mathbf{x}$ if there exists $\mathbf{x}'$ such that $\mathbf{x} \xrightarrow{a} \mathbf{x}'$. We denote the components of a SHIOA $\mathcal{A}$ by $X_{\mathcal{A}}, Y_{\mathcal{A}}$ etc.

An execution of $\mathcal{A}$ records the valuations of all its variables and the occurrences of all actions over a particular run. An execution is *closed* if it is finite and the last trajectory in it is closed.

An *execution fragment* of $\mathcal{A}$ is a finite or infinite sequence $\alpha = \tau_0 a_1 \tau_1 a_2 \ldots$, such that for all $i$ in the sequence, $a_i \in A$, $\tau \in trajs(\mathcal{S})$ for some $\mathcal{S} \in \mathscr{S}$, and $\tau_i.\mathsf{lstate} \xrightarrow{a_{i+1}} \tau_{i+1}.\mathsf{fstate}$. An execution fragment is an *execution* if $\tau_0.\mathsf{fstate} \in Q_0$. The first state of $\alpha$, $\alpha.\mathsf{fstate}$, is $\tau_0.\mathsf{fstate}$, and for a closed $\alpha$, its last state, $\alpha.\mathsf{lstate}$, is the last state of its last trajectory. The *limit time* of $\alpha$, $\alpha.\mathsf{ltime}$, is defined to be $\sum_i \tau_i.\mathsf{ltime}$. The set of executions and reachable states of $\mathcal{A}$ are denoted by $\mathsf{Execs}_{\mathcal{A}}$ and $\mathsf{Reach}_{\mathcal{A}}$. A set of states $I \subseteq Q$ is said to be an *invariant* of $\mathcal{A}$ iff $\mathsf{Reach}_{\mathcal{A}} \subseteq I$.

## 3. PERIODICALLY CONTROLLED HYBRID SYSTEMS

In this section, we define a subclass of SHIOAs that is suitable for modeling sampled control systems and embedded systems with periodic sensing and actuation. The main result of this section, Theorem 3.4, gives a sufficient condition for proving invariant properties of this subclass.

### 3.1 Periodically Controlled Hybrid I/O Automata

A *Periodically Controlled Hybrid Automaton (PCHA)* is an SHIOA with a set of (*control*) actions that occur roughly periodically. These *control* actions alter the actual control signal (input) that feeds to the plant and may change the continuous and the discrete state variables of the automaton. The automaton may have other actions that change only the discrete state of the automaton. These actions can model, for example, sensor inputs and the change in the set-point of the controller from higher-level inputs. For the sake of simplicity, we consider the PCHAs of the form shown in Figure 1, however, Theorem 3.4 generalizes to PCHAs with other input, output, and internal actions.

Let $\mathcal{X} \subseteq \mathbb{R}^n$, for some $n \in \mathbb{N}$, and $\mathcal{L}, \mathcal{Z}$, and $\mathcal{U}$ be arbitrary types. Four key variables of PCHA $\mathcal{A}$ are

(a) *continuous state* variable $s$ of type $\mathcal{X}$, initialized to $s_0$,

(b) *discrete state* (*location* or *mode*) variable *loc* of type $\mathcal{L}$, initialized to $l_0$,

(c) *command* variable $z$ of type $\mathcal{Z}$, initialized to $z_0$, and

(d) *control* variable $u$ of type $\mathcal{U}$, initialized to $u_0$.

The continuous state generally includes the continuous state of the plant and some internal state of the controller. The discrete state represents the mode of the system. The command variable is used to store externally produced input commands or sensor updates. The control variable stores the control input to the plant. Finally, the *now* and *next* variables are used for triggering the control action periodically.

PCHA $\mathcal{A}$ has two types of actions: (a) through input action update $\mathcal{A}$ learns about new externally produced input commands such as set-points, waypoints. When an update($z'$) action occurs, $z'$ is recorded in the command variable $z$. (b) The control action changes the control variable $u$. This action occurs roughly periodically starting from time 0; the time gap between two successive occurrences is within $[\Delta_1, \Delta_1 + \Delta_2]$ where $\Delta_1 > 0$ and $\Delta_2 \geq 0$. When control occurs, $loc$ and $s$ are computed as a function of their current values and that of $z$, and $u$ is computed as a function of the new values of $loc$ and $s$.

For each value of $l \in \mathcal{L}$, the continuous state $s$ evolves according to the trajectories specified by state model $smodel(l)$. That is, $s$ evolves according to the differential equation $\dot{s} = f_l(s, u)$. The timing of control behavior is enforced by the precondition of control and the stopping condition of the state models.

```
signature                                    1   internal control                          16
  internal control                               pre now ≥ next
  input update(z' : Z)                       3   eff next := now + Δ1;                     18
                                                   ⟨loc, s⟩ := h(loc, s, z);
variables                                    5     u := g(loc, s)                          20
  internal s : X := s0
  internal discrete loc : L := l0,           7 trajectories                              22
    z : Z := z0, u : U := u0                     trajdef smodel(l : L)
  internal now : R≥0 := 0,                   9     invariant loc = l                     24
    next : R := −Δ2                              evolve d(now) = 1; d(s) = fl(s, u)
                                             11      stop when now = next + Δ2           26
transitions
  input update(z')                           13
  eff z := z'
```

Fig. 1. PHCA with parameters $\Delta_1$, $\Delta_2$, $g$, $h$, $\{f_l\}_{l \in \mathcal{L}}$. See, for example, [Mitra 2007] for the description of the language.

### 3.2 Invariant Verification

Proving invariant properties of hybrid automata is a central problem in formal verification. Invariants are used for overapproximating the reachable states of a given system, and therefore, can be used for verifying safety properties.

Given a candidate invariant set $\mathcal{I} \subseteq Q$, we are interested in verifying that $\mathsf{Reach}_A \subseteq \mathcal{I}$. For continuous dynamical systems, checking the well-known subtangential condition (see, for example [Bhatia and Szegö 1967]) provides a sufficient condition for proving invariance of a set $\mathcal{I}$ that is bounded by a closed surface. Theorem 3.4 provides an analogous sufficient condition for PCHAs. In general, however, invariant sets $\mathcal{I}$ for PCHAs have to be defined by a collection of functions

instead of a single function. For each mode $l \in \mathcal{L}$, we assume that the invariant set $I_l \subseteq \mathcal{X}$ for the continuous state is defined by a collection of $m$ *boundary functions* $\{F_{lk}\}_{k=1}^{m}$, where $m$ is some natural number and each $F_{lk} : \mathcal{X} \to \mathbb{R}$ is a differentiable function[3]. Formally,

$$I_l \triangleq \{s \in \mathcal{X} \mid \forall k \in \{1, \ldots, m\}, F_{lk}(s) \geq 0\} \text{ and } \mathcal{I} \triangleq \{\mathbf{x} \in Q | \ \mathbf{x}.s \in I_{\mathbf{x}.loc}\}.$$

Note that the overall candidate invariant set $\mathcal{I}$ does not restrict the values of the command or the control variables. In the remainder of this section, we develop a set of sufficient conditions for checking that $\mathcal{I}$ is indeed an invariant of a given PCHA. Lemma 3.1 modifies the standard inductive technique for proving invariance, so that it suffices to check invariance with respect to Control transitions and Control-free execution fragments of length not greater than $\Delta_1 + \Delta_2$.

**Lemma 3.1.** *Suppose $Q_0 \subseteq \mathcal{I}$ and the following two conditions hold:*

*(a) (Control steps) For each state $\mathbf{x}, \mathbf{x}' \in Q$, if $\mathbf{x} \overset{\mathsf{control}}{\rightarrow} \mathbf{x}'$ and $\mathbf{x} \in \mathcal{I}$ then $\mathbf{x}' \in \mathcal{I}$.*

*(b) (Control-free fragments) For each closed execution fragment $\beta = \tau_0$ update$(z_1)$ $\tau_1$ update$(z_2)$ $\ldots$ $\tau_n$ starting from a state $\mathbf{x} \in \mathcal{I}$ where each $z_i \in \mathcal{Z}$, if $\mathbf{x}.next - \mathbf{x}.now = \Delta_1$ and $\beta.\mathsf{ltime} \leq \Delta_1 + \Delta_2$, then $\beta.\mathsf{lstate} \in \mathcal{I}$.*

*Then* $\mathsf{Reach}_{\mathcal{A}} \subseteq \mathcal{I}$.

Proof. Consider any reachable state $\mathbf{x}$ of $\mathcal{A}$ and any execution $\alpha$ such that $\alpha.\mathsf{lstate} = \mathbf{x}$. We can write $\alpha$ as $\beta_0$ control $\beta_1$ control $\ldots \beta_k$, where each $\beta_i$ is control-free execution fragment of $\mathcal{A}$, i.e., execution fragments in which only update actions occur. From condition (a), it follows that for each $i \in \{0, \ldots, k\}$, if $\beta_i.\mathsf{lstate} \in \mathcal{I}$, then $\beta_{i+1}.\mathsf{fstate} \in \mathcal{I}$.

Thus, it suffices to prove that for each $i \in \{0, \ldots, k\}$, if $\beta_i.\mathsf{fstate} \in \mathcal{I}$, then $\beta_i.\mathsf{lstate} \in \mathcal{I}$. We fix an $i \in \{0, \ldots, k\}$ and assume that $\beta_i.\mathsf{fstate} \in \mathcal{I}$. Let $\beta_i = \tau_0$ update$(z_1)$ $\tau_1$ update$(z_2) \ldots \tau_n$, where for $j \in \{0, \ldots, n\}$, $z_j \in \mathcal{Z}$ and $\tau_j$ is a trajectory of $\mathcal{A}$. If $i = 0$, then $\beta_i.\mathsf{ltime} = 0$ and $\beta_i.\mathsf{lstate} \lceil \{loc, s\} = \beta_i.\mathsf{fstate} \lceil \{loc, s\}$ since the first control action occurs at time 0 and update transitions do not affect the value of $loc$ and $s$. Therefore, $\beta_i.\mathsf{lstate} \in \mathcal{I}$. Otherwise, $i > 0$ and since $\beta_i$ starts immediately after a control action, $\beta.\mathsf{fstate} \lceil next - \beta.\mathsf{fstate} \lceil now = \Delta_1$. From periodicity of main actions, we know that $\beta_i.\mathsf{ltime} \leq \Delta_1 + \Delta_2$, and hence from condition (b) it follows that $\beta_i.\mathsf{lstate} \in \mathcal{I}$. □

Invariance of control steps can often be checked through case analysis which can be partially automated using a theorem prover [Owre et al. 1996]. The next key lemma provides a sufficient condition for proving invariance of control-free fragments. Since, control-free fragments do not change the valuation of the $loc$ variable, for this part, we fix a value $l \in \mathcal{L}$. For each index of the boundary functions $j \in \{1, \ldots m\}$, we define the set $\partial I_j$ to be part of the set $I_l$ where the function $F_{lj}$ vanishes. That is, $\partial I_j \triangleq \{x \in \mathcal{X} \mid F_{lj}(x) = 0\}$. For the sake of brevity, we call $\partial I_j$ the $j^{th}$ boundary of $I_l$ even though strictly speaking, the $j^{th}$ boundary of

[3]Identical size $m$ of the collections simplifies our notation; different number of boundary functions for different values of $l$ can be handled by extending the theorem in an obvious way.

$I_l$ is only a subset of $\partial I_j$ according to the standard topological definition. Similarly, we say that the *boundary* of $I_l$, is $\partial I_l = \bigcup_{j\in\{1,\ldots,m\}} \partial I_j$.

**Lemma 3.2.** *Suppose that there exists a collection $\{C_j\}_{j=1}^m$ of subsets of $I_l$ such that the following conditions hold:*

*(a) (Subtangential) For each $s_0 \in I_l \setminus C_j$ and $s \in \partial I_j$, $\frac{\partial F_{lj}(s)}{\partial s} \cdot f_l(s, g(l, s_0)) \geq 0$.*
*(b) (Bounded distance) $\exists\ c_j > 0$ such that $\forall\ s_0 \in C_j, s \in \partial I_j$, $||s - s_0|| \geq c_j$.*
*(c) (Bounded speed) $\exists\ b_j > 0$ such that $\forall\ s_0 \in C_j, s \in I_l$, $||f_l(s, g(l, s_0))|| \leq b_j$,*
*(d) (Fast sampling) $\Delta_1 + \Delta_2 \leq \min_{j\in\{1,\ldots,m\}} \frac{c_j}{b_j}$.*

*Then, any* control*-free execution fragment $\beta$, with $\beta.\mathsf{ltime} \leq \Delta_1 + \Delta_2$, starting from a state in $I_l$ where $next - now = \Delta_1$, remains within $I_l$.*

In Figure 2, the control and control-free fragments are shown by bullets and lines, respectively. A fragment starting in $\mathcal{I}$ and leaving $\mathcal{I}$, must cross $\partial I_1$ or $\partial I_2$. Consider the following four cases.

(1) If $u$ is evaluated outside both $C_1$ and $C_2$ (e.g. $\tau_2$, $\tau_4$ and $\tau_6$), then condition (a) guarantees that the fragment does not cross $\partial I_j$ where $j \in \{1, 2\}$ because when it reaches $\partial I_j$, the vector field governing its evolution points inwards with respect to $\partial I_j$.
(2) If $u$ is evaluated inside $C_1$ but outside $C_2$ (e.g. $\tau_1$ and $\tau_7$), then by the previous reasoning, condition (a) guarantees that the fragment does not cross $\partial I_2$. In addition, conditions (b) and (c) guarantee that it takes finite time before the fragment reaches $\partial I_1$ and condition (d) guarantees that this finite time is at least $\Delta_1 + \Delta_2$; thus, before the fragment crosses $\partial I_1$, $u$ is evaluated again.
(3) If $u$ is evaluated outside $C_1$ but inside $C_2$ (e.g. $\tau_3$), then by a symmetric argument, the fragment does not cross $\partial I_1$ or $\partial I_2$.
(4) If $u$ is evaluated inside both $C_1$ and $C_2$ (e.g. $\tau_5$), then conditions (b), (c) and (d) guarantee that $u$ is evaluated again before fragment crosses $\partial I_1$ or $\partial I_2$.

PROOF. We fix a control-free execution fragment $\beta = \tau_0\mathsf{update}(z_1)\tau_1\mathsf{update}(z_2)\ldots\tau_n$ such that at $\beta.\mathsf{fstate}$, $next - now = \Delta_1$. Without loss of generality we assume that at $\beta.\mathsf{fstate}$, $z = z_1$, $loc = l$, and $s = x_1$, where $z_1 \in \mathcal{Z}, l \in \mathcal{L}$ and $x_1 \in I_l$. We have to show that at $\beta.\mathsf{lstate}$, $s \in I_l$.

First, observe that for each $k \in \{0, \ldots, n\}$, $(\tau_k \downarrow s)$ is a solution of the differential equation(s) $d(s) = f_l(s, g(l, x_1))$. Let $\tau$ be the pasted trajectory $\tau_0 \frown \tau_1 \frown \ldots \tau_n$.[4] Let $\tau.\mathsf{ltime}$ be $T$. Since the update action does not change $s$, $\tau_k.\mathsf{lstate} \lceil s = \tau_{k+1}.\mathsf{fstate} \lceil s$ for each $k \in \{0, \ldots, n-1\}$. As the differential equations are time invariant, $(\tau \downarrow s)$ is a solution of $d(s) = f_l(s, g(l, x_1))$. We define the function $\gamma : [0, T] \to \mathcal{X}$ as $\forall\ t \in [0, T]$, $\gamma(t) \triangleq (\tau \downarrow s)(t)$. We have to show that $\gamma(T) \in I_l$. Suppose, for the sake of contradiction, that there exists $t^* \in [0, T]$, such that $\gamma(t^*) \notin I_l$. By the definition of $I_l$, there exists $i$ such that $F_{li}(\gamma(0)) \geq 0$ and $F_{li}(\gamma(t^*)) < 0$. We pick one such $i$ and fix it for the remainder of the proof. Since $F_{li}$ and $\gamma$ are continuous, from intermediate value theorem, we know that there exists a time $t_1$

[4] $\tau_1 \frown \tau_2$ is the trajectory obtained by concatenating $\tau_2$ at the end of $\tau_1$.

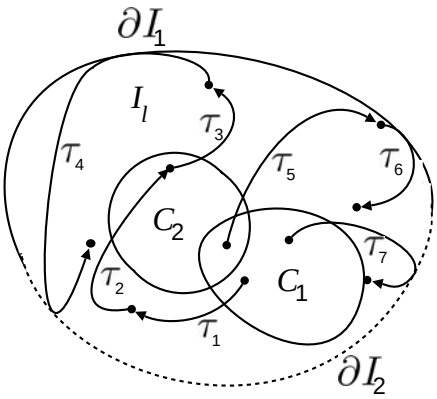



Fig. 2. A graphical explanation of Lemma 3.2 showing an invariant set $I_l$ defined by two boundary functions. The boundary $\partial I_1$ is drawn in solid line whereas the boundary $\partial I_2$ is drawn in dotted line. The corresponding sets $C_1$ and $C_2$ are also shown.

before $t^*$ where $F_{li}$ vanishes and that there is some finite time $\epsilon > 0$ after $t_1$ when $F_{li}$ is strictly negative. Formally, there exists $t_1 \in [0, t^*)$ and $\epsilon > 0$ such that for all $t \in [0, t_1]$, $F_{li}(\gamma(t)) \geq 0$, $F_{li}(\gamma(t_1)) = 0$, and for all $\delta \in (0, \epsilon]$, $F_{li}(\gamma(t_1 + \delta)) < 0$.

Case 1: $x_1 \in I_l \setminus C_i$. Since $F_{li}(\gamma(t_1)) = 0$, by definition, $\gamma(t_1) \in \partial I_i$. But from the value of $F_{li}(\gamma(t))$ where $t$ is near to $t_1$, we get that $\frac{\partial F_{li}}{\partial t}(t_1) = \frac{\partial F_{li}}{\partial s}(\gamma(t_1)) \cdot f_l(\gamma(t_1), g(l, x_1)) < 0$. This contradicts condition (a).

Case 2: $x_1 \in C_i$. Since for all $t \in [0, t_1]$, $F_{li}(\gamma(t)) \geq 0$ and $F_{li}(\gamma(t_1)) = 0$, we get that for all $t \in [0, t_1]$, $\gamma(t) \in I_l$ and $\gamma(t_1) \in \partial I_i$. So from condition (b) and (c), we get $c_i \leq \|\gamma(t_1) - x_1\| = \left\| \int_0^{t_1} f_l(\gamma(t), g(l, x_1))dt \right\| \leq b_i t_1$. That is, $t_1 \geq \frac{c_i}{b_i}$. But we know that $t_1 < t^* \leq T$ and periodicity of Control actions $T \leq \Delta_1 + \Delta_2$. Combining these, we get $\Delta_1 + \Delta_2 > \frac{c_i}{b_i}$ which contradicts condition (d). □

For PCHAs with certain properties, the following lemma provides sufficient conditions for the existence of the bounds $b_j$ and $c_j$ which satisfy the bounded distance and bounded speed conditions of Lemma 3.2.

**Lemma 3.3.** *For a given $l \in L$, let $U_l = \{g(l, s) \mid l \in \mathcal{L}, s \in I_l\} \subseteq \mathcal{U}$ and suppose $I_l$ is compact and $f_l$ is continuous in $I_l \times U_l$. The bounded distance and bounded speed conditions (of Lemma 3.2) are satisfied if $C_j \subset I_l$ satisfies the following conditions:*

$$C_j \text{ is closed} \tag{1}$$
$$C_j \cap \partial I_j = \emptyset \tag{2}$$

Proof. From the continuity of $F_{lj}$, we can assume, without loss of generality, that $\partial I_j \neq \emptyset$. This is because if $\partial I_j = \emptyset$, then for all $s \in \mathcal{X}$, it must be either $F_{lj}(s) > 0$ or $F_{lj}(s) < 0$, that is, $F_{lj}$ is not needed to describe $I_l$. In addition, the case where $C_j = \emptyset$ is trivial since conditions (b) and (c) of Lemma 3.2 are satisfied for any arbitrary large $c_j$ and arbitrary small $b_j$. So for the rest of the proof, we assume that $\partial I_j \neq \emptyset$ and $C_j \neq \emptyset$. Since $I_l$ is compact and $C_j$ and $\partial I_j$ are closed,

$C_j$ and $\partial I_j$ are also compact. Consider a function $G_j : \partial I_j \to \mathbb{R}$ defined by

$$G_j(s) = \min_{s_0 \in C_j} \|s - s_0\|,$$

where $\|\cdot\|$ is a norm on $\mathbb{R}^n$. Due to the continuity of $\|\cdot\|$ and the compactness and nonemptyness of $C_j$, $G_j$ is continuous and since $C_j \cap \partial I_j = \emptyset$, we get that for all $s \in \partial I_j, G_j(s) > 0$. Since $\partial I_j$ is compact and nonempty, $G_j$ attains its minimum in $\partial I_j$. So there exists $c_j > 0$ such that $\min_{s \in \partial I_j} G_j(s) \geq c_j$.

Next, consider a function $H_j : I_l \to \mathbb{R}$ defined by

$$H_j(s) = \max_{s_0 \in C_j} \|f_l(s, g(l, s_0))\|.$$

Using the continuity of $f_l$, the compactness and nonemptyness of $C_j$ and $I_l$ and the same argument as above, we can conclude that there exists $b_j \geq 0$ such that $\max_{s \in I_l} H_j(s) \leq b_j$. □

Theorem 3.4 combines the above lemmas and provides sufficient conditions for invariance of $\mathcal{I}$.

**Theorem 3.4.** *Consider a* PCHA $\mathcal{A}$ *and a set* $\mathcal{I} \subseteq Q_{\mathcal{A}}$. *Suppose* $Q_{0\mathcal{A}} \subseteq \mathcal{I}$, $\mathcal{A}$ *satisfies control invariance condition of Lemma 3.1, and conditions (a)–(d) of Lemma 3.2 for each* $l \in \mathcal{L}_{\mathcal{A}}$. *Then* $\mathsf{Reach}_{\mathcal{A}} \subseteq \mathcal{I}$.

Proof. The proof follows directly from Lemma 3.1 and Lemma 3.2 since if conditions (a)–(d) of Lemma 3.2 are satisfied for any $l \in \mathcal{L}$, then condition (b) of Lemma 3.1 is satisfied. □

Although the PCHA of Figure 1 has one action of each type, Theorem 3.4 can be extended for periodically controlled hybrid systems with arbitrary number of input and internal actions. For PCHAs with polynomial vector-fields, given the semi-algebraic sets $I_l$ and $C_j$, checking condition (a) and finding $c_j$ and $b_j$ that satisfy conditions (b) and (c) of Lemma 3.2 can be formulated as a sum of squares optimization problem (provided that $I_l$ and $C_j$ are basic semi-algebraic sets) or proving emptiness of some certain semi-algebraic sets based on quantifier elimination. The sum of squares formulation is presented in the next section and allows the proof to be automated using, for example, SOSTOOLS [Prajna et al. 2002]. The quantifier elimination problem can also be formulated and allows the proof to be automated using, for example, QEPCAD [Brown 2003]. Alternatively, in Section 3.4, we show how an invariant set can be automatically computed using the constraint-based approach presented in [Gulwani and Tiwari 2008].

### 3.3 Sum of Squares Formulation for Checking the Invariant Conditions

Suppose the candidate invariant set $I_l$ is a basic semi-algebraic set, i.e., each of the boundary functions $F_{lk} : \mathcal{X} \to \mathbb{R}$ is a real polynomial. This section presents a sum of squares formulation for the following two cases: (1) checking condition (a) and finding the $c_j$ and $b_j$ that satisfy conditions (b) and (c) of Lemma 3.2 for a given basic semi-algebraic subset $C_j$, and (2) finding a subset $C_j$ such that conditions (a)–(c) of Lemma 3.2 are satisfied. For the first case, the sum of squares problem is convex and can be solved using, for example, SOSTOOLS [Prajna et al. 2002]. For the second case, however, the problem is not convex but can still be automatically solved using an iterative scheme as presented in [Prajna and Jadbabaie 2004].

### Checking Invariant Condition for a Given Subset

Suppose $C_j$ a basic semi-algebraic set, that is, there exists a natural number $p$ such that $C_j$ can be written as

$$C_j = \{s \in I_l \mid \forall i \in \{1, \ldots, p\}, G_{ji}(s) \geq 0\} \tag{3}$$

where $G_{ji} : \mathcal{X} \to \mathbb{R}$ is a real polynomial for each $i \in \{1, \ldots, p\}$. Then the set $I_l \setminus C_j = I_l \cap \overline{C}_j$ is given by

$$\begin{aligned} I_l \setminus C_j = \{s \in \mathcal{X} \mid\ & (F_{l1}(s) \geq 0 \cap \ldots \cap F_{lm} \geq 0 \cap G_{j1} < 0) \ \cup \\ & (F_{l1}(s) \geq 0 \cap \ldots \cap F_{lm} \geq 0 \cap G_{j2} < 0) \ \cup \ldots \cup \\ & (F_{l1}(s) \geq 0 \cap \ldots \cap F_{lm} \geq 0 \cap G_{jp} < 0)\} \end{aligned} \tag{4}$$

The following provides a sufficient condition for condition (a) of Lemma 3.2.

For each $k \in \{1, \ldots, p\}$, there exist sums of squares $\mu_k(s)$, $\rho_{k,i}(s)$ and $\sigma_{k,i}(s)$ for $i \in \{1, \ldots, m\}$ and a polynomial $\nu_k(s)$ such that

$$\frac{\partial F_{lj}(s)}{\partial s} \cdot f_l(s, g(l, s_0)) - \sum_{i=1}^{m} \rho_{k,i}(s) F_{li}(s) - \nu_k(s) F_{lj}(s) - \sum_{i=1}^{m} \sigma_{k,i}(s_0) F_{li}(s_0) + \mu_k(s_0) G_{jk}(s_0)$$

is a sum of squares.

Condition (b) of Lemma 3.2 can be formulated as the following optimization problem.

Minimize $-c_j$ such that there exist sums of squares $\gamma_i(s)$ for $i \in \{1, \ldots, m\}$ and $\lambda_i(s)$ for $i \in \{1, \ldots p\}$ and a polynomial $\gamma_{m+1}(s)$ such that

$$\|s - s_0\|^2 - c_j^2 - \sum_{i=1}^{m} \gamma_i(s) F_{li}(s) - \gamma_{m+1}(s) F_{lj}(s) - \sum_{i=1}^{p} \lambda_i(s_0) G_{ji}(s_0)$$

is a sum of squares.

Finally, condition (c) of Lemma 3.2 can be formulated as the following optimization problem.

Minimize $b_j$ such that there exist sums of squares $\zeta_i(s)$ for $i \in \{1, \ldots, m\}$ and $\eta_i(s)$ for $i \in \{1, \ldots p\}$ such that

$$b_j^2 - \|f_l(s, g(l, s_0)\|^2 - \sum_{i=1}^{m} \zeta_i(s) F_{li}(s) - \sum_{i=1}^{p} \eta_i(s_0) G_{ji}(s_0)$$

is a sum of squares.

### Finding a Subset for Checking the Invariant Conditions

Suppose $C_j = \{s \in I_l \mid G_j(s) \geq 0\}$. In this case, we only have to find a polynomial $G_j(s)$. This problem can be formulated as follows: Find sums of squares $\eta_1(s), \ldots, \eta_4(s)$, $\rho_i(s)$, $\sigma_i(s)$, $\gamma_i(s)$, $\kappa_i(s)$ and $\zeta_i(s)$ for $i \in \{1, \ldots, m\}$ and polynomials $G_j(s)$, $\nu(s)$ and $\gamma_{m+1}(s)$ such that the followings are sums of squares

(a) $F_{lj}(s) - \eta_1(s) G_j(s)$

(b) $\frac{\partial F_{lj}(s)}{\partial s} \cdot f_l(s, g(l, s_0)) - \sum_{i=1}^{m} \rho_i(s) F_{li}(s) - \nu(s) F_{lj}(s) - \sum_{i=1}^{m} \sigma_i(s_0) F_{li}(s_0) + \eta_2(s_0) G_j(s_0)$,

(c) $\|s-s_0\|^2 - c_j^2 - \sum_{i=1}^{m} \gamma_i(s)F_{li}(s) - \gamma_{m+1}(s)F_{lj}(s) - \sum_{i=1}^{m} \kappa_i(s_0)F_{li}(s_0) - \eta_3(s_0)G_j(s_0)$, and

(d) $b_j^2 - \|f_l(s, g(l, s_0)\|^2 - \sum_{i=1}^{m} \zeta_i(s)F_{li}(s) - \eta_4(s_0)G_j(s_0)$.

### 3.4 Example

Consider a one-dimensional system whose the continuous state needs to be regulated such that it stays within a certain safety region. The system has the following variables:

(a) a continuous state variable $s$ of type $\mathbb{R}$, initialized to $s_0 \in [D-\delta, D+\delta]$ where $D \in \mathbb{R}$ is a system parameter and $\delta \in \mathbb{R}_{\geq 0}$ is an arbitrary uncertainty in the initial state of the system,

(b) a discrete state variable $loc$ of type $\mathcal{L} = \{0, 1\}$,

(c) a control variable $u$ of type $\mathcal{U} = \{a_1, a_2\}$ where $a_1 \in \mathbb{R}_-$ and $a_2 \in \mathbb{R}_+$ are system parameters.

Figure 3 shows the SHIOA specification of this state regulator system. The control action occurs once every $\Delta$ time starting from time 0 where $\Delta \in \mathbb{R}_+$. This action updates the values of the variables $loc$ and $u$ based on the system parameter $D$ as follows.

A. If $s > D$, then $loc$ is set to 1 (line 16). Otherwise, $loc$ is set to 0 (line 17). That is, the function $h$ of line 19 of Figure 1 which updates $loc$ and $s$ is defined as $h = \langle h_l, h_s \rangle$ where $h_l$ and $h_s$ describe the discrete transition of $loc$ and $s$ respectively and

$$h_s(loc, s, z) = s, \tag{5}$$

$$h_l(loc, s, z) = \begin{cases} 0 & \text{if } s \leq D \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

B. Based on the updated value of $loc$, $u$ is computed using function $g$ of line 20 of Figure 1 which is defined as follows (lines 18–19):

$$g(loc, s) = \begin{cases} a_1 & \text{if } loc = 1 \\ a_2 & \text{otherwise} \end{cases} \tag{7}$$

Along a trajectory, the continuous state $s$ evolves according to the differential equation $\dot{s} = u$ (line 22). That is, for any $l \in \mathcal{L}$, the function $f_l$ of line 25 of Figure 1 is defined as $f_l(s, u) = u$.

*Invariant.* For each mode $l \in \mathcal{L}$, we let $I_l = [D - \max(\delta, -a_1\Delta), D + \max(\delta, a_2\Delta)]$. That is, the candidate invariant set $I_l$ is defined by two boundary functions

$$F_{l1}(s) = s - D + \max(\delta, -a_1\Delta), \text{ and } F_{l2}(s) = -s + D + \max(\delta, a_2\Delta). \tag{8}$$

The overall candidate invariant set is then given by $\mathcal{I} \triangleq \{\mathbf{x} \in Q |\ F_{l1}(\mathbf{x}.s) \geq 0 \text{ and } F_{l2}(\mathbf{x}.s) \geq 0\}$.

```
signature                                          1
  internal control
  input update(z' : Z)                             3

variables                                          5
  internal s : ℝ := s_0 ∈ [D − δ, D + δ]
  internal discrete loc : {0, 1},                  7
    u : {a_1, a_2}
  internal now : ℝ_≥0 := 0,                        9
    next : ℝ_≥0 := 0

transitions                                        12
  internal control
  pre now ≥ next                                   14
  eff next := now + Δ;
    if s > D then loc := 1                         16
    else loc := 0 fi
    if loc = 1 then u := a_1                       18
    else u := a_2 fi
                                                   20
trajectories
  evolve d(now) = 1; d(s) = u                      22
  stop when now = next
```

Fig. 3. The state regulator system with parameters $a_1 \in \mathbb{R}_-$, $a_2 \in \mathbb{R}_+$, $\Delta \in \mathbb{R}_+$, $\delta \in \mathbb{R}_{\geq 0}$ and $D \in \mathbb{R}$.

*Proving Invariant.* We use Theorem 3.4 to show that $\mathcal{I}$ is in fact an invariant of the system. Clearly, the initial state is contained in $\mathcal{I}$ and the control invariance condition of Lemma 3.1 is satisfied since control actions do not change the value of $s$. Thus, we only need to show that there exist subsets $C_1$ and $C_2$ of $I_l$ such that conditions (a)–(d) of Lemma 3.2 are satisfied. It can be easily verified that with $C_1 = [D, D + \max(\delta, a_2\Delta)]$ and $C_2 = [D - \max(\delta, -a_1\Delta), D]$, we get $c_1 = \max(\delta, -a_1\Delta)$, $c_2 = \max(\delta, a_2\Delta)$, $b_1 = -a_1$, $b_2 = a_2$, and conditions (a)–(d) of Lemma 3.2 are satisfied.

*Automatically Finding an Invariant.* We consider the case where $a_1 = -1$ and $a_2 = 1$. Assume that an invariant $I_l$ for both modes $l = 0$ and $l = 1$ has the following form: $I_l = \{s \in \mathbb{R} \mid F_{l1}(s) \geq 0 \text{ and } F_{l2}(s) \geq 0\}$ where $F_{l1}(s) = s - \eta_1$, $F_{l2}(s) = -s + \eta_2$ and $\eta_1 \geq D - \delta$ and $\eta_2 \geq D + \delta$ are constants that need to be computed such that all the conditions of Lemma 3.2 are satisfied.

To prove that $I_l$ is in fact an invariant, we use the sets $C_1$ and $C_2$ of the following forms: $C_1 = \{s \in \mathbb{R} \mid G_1(s) \geq 0 \text{ and } F_{l2}(s) \geq 0\}$ and $C_2 = \{s \in \mathbb{R} \mid F_{l1}(s) \geq 0 \text{ and } G_2(s) \geq 0\}$ where $G_1(s) = s - \kappa_1$, $G_2(s) = -s + \kappa_2$ and $\kappa_1$ and $\kappa_2$ are constants to be determined.

Clearly, for any $s, s_0 \in \mathbb{R}$ and $l \in \mathcal{L}$, $\|f_l(s, g(l, s_0))\| = \|g(l, s_0)\| = 1$. Thus, condition (c) of Lemma 3.2 is satisfied with $b_j = 1$ for any sets $C_j$ and $I_l$. With the particular form of the sets $C_1$, $C_2$ and $I_l$ we have previously chosen, it can be easily checked that the problem of finding $\eta_1$, $\eta_2$, $\kappa_1$ and $\kappa_2$ such that all the conditions of Lemma 3.2 are satisfied for $j = 1$ is equivalent to finding $\eta_1$, $\eta_2$, $\kappa_1$ and $\kappa_2$ such that for all $s, s_0 \in \mathbb{R}$, the followings are satisfied:

(a) $(F_{l1}(s_0) < 0) \vee (F_{l2}(s_0) < 0) \vee (G_1(s_0) \geq 0) \vee (F_{l1}(s) \neq 0) \vee (F_{l2}(s) < 0) \vee (s_0 \leq D)$

(b) $\kappa_1 \leq \eta_2$

(c) $\kappa_1 > \eta_1$

(d) $\kappa_1 - \eta_1 \geq \Delta$

Similarly, for $j = 2$, the following conditions need to be satisfied for all $s, s_0 \in \mathbb{R}$:

(e) $(F_{l1}(s_0) < 0) \vee (F_{l2}(s_0) < 0) \vee (G_2(s_0) \geq 0) \vee (F_{l1}(s) < 0) \vee (F_{l2}(s) \neq 0) \vee (s_0 > D)$

(f) $\kappa_2 \geq \eta_1$

(g) $\kappa_2 < \eta_2$

(h) $\eta_2 - \kappa_2 \geq \Delta$

As described in [Gulwani and Tiwari 2008], the validity of condition (a) can be proved by finding a constant $\lambda_1$ and non-negative constants $\nu_1, \ldots, \nu_3$ and $\mu_1, \ldots, \mu_3$ such that

$$\nu_1 F_{l1}(s_0) + \nu_2 F_{l2}(s_0) - \mu_1 G_1(s_0) + \lambda_1 F_{l1}(s) + \nu_3 F_{l2}(s) + \mu_2(s_0 - D) + \mu_3 = 0 \quad (9)$$

and at least one of the $\mu_1, \mu_2, \mu_3$ is strictly positive. Similarly, the validity of condition (e) can be proved by finding a constant $\lambda_2$ and non-negative constants $\nu_4, \ldots, \nu_7$ and $\mu_4, \mu_5$ such that

$$\nu_4 F_{l1}(s_0) + \nu_5 F_{l2}(s_0) - \mu_4 G_2(s_0) + \nu_6 F_{l1}(s) + \lambda_2 F_{l2}(s) + \nu_7(D - s_0) + \mu_5 = 0 \quad (10)$$

and either $\mu_4 > 0$ or $\mu_5 > 0$ (or both).

Using the tool presented in [Gulwani and Tiwari 2008], the unknowns that satisfy (9), (10) and conditions (b)–(d) and (f)–(h) are found for $D = 1$, $\delta = 0.1$ and $\Delta = 0.1$ to be: $\eta_1 = 0.8$, $\eta_2 = 1.2$, $\kappa_1 = 0.9$, $\kappa_2 = 1.1$, $\nu_1 = 1$, $\nu_2 = 2$, $\mu_1 = 16$, $\lambda_1 = 0$, $\nu_3 = 0$, $\mu_2 = 17$, $\mu_3 = 1$, $\nu_4 = 0$, $\nu_5 = 0$, $\mu_4 = 20$, $\nu_6 = 0$, $\lambda_2 = 0$, $\nu_7 = 20$ and $\mu_5 = 2$. That is, the invariant set is given by $I_l = [0.8, 1.2]$ (whereas the invariant set we have verified manually is given by $I_l = [0.9, 1.1]$).

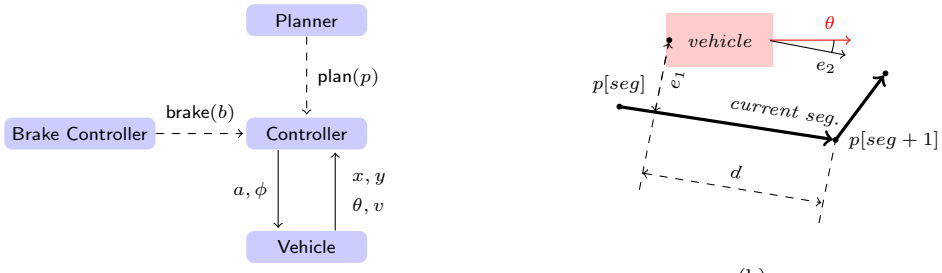
## 4. AUTONOMOUS VEHICLE SYSTEM

In this section, we describe a subsystem of an autonomous ground vehicle (Alice) consisting of the physical vehicle and the controller (see, Figure 4(a)). Vehicle captures its the position, orientation, and the velocity of the vehicle on the plane. Controller receives information about the state of the vehicle and periodically computes the input steering ($\phi$) and the acceleration ($a$). Controller also receives an infinite[5] sequence of waypoints from a Planner and its objective is to compute $a$ and $\phi$ such that the vehicle (a) remains within a certain bounded distance $e_{max}$ of the planned path, and (b) makes progress towards successive waypoints at a target speed. Property (a) together with the assumption (possibly guaranteed by Planner) that all planned paths are at least $e_{max}$ distance away from obstacles, imply that the Vehicle does not collide with obstacles. While the Vehicle makes progress towards a certain waypoint, the subsequent waypoints may change owing to the discovery of new obstacles, short-cuts, and changes in the mission plan. Finally, the Controller may receive an externally triggered brake input, to which it must react by slowing the vehicle down.

### 4.1 Vehicle

The Vehicle automaton of Figure 4 specifies the dynamics of the autonomous ground vehicle with acceleration ($a$) and steering angle ($\phi$) as inputs. It has two parameters: (a) $\phi_{max} \in (0, \frac{\pi}{2}]$ is the physical limit on the steering angle, and (b) $L$ is the wheelbase. The main output variables of Vehicle are (a) $x$ and $y$ coordinates of the vehicle with respect to a global coordinate system, (b) orientation $\theta$ of the vehicle

[5]The verification technique can be extended in an obvious way to handle the case where the vehicle has to follow a finite sequence of waypoints and halt at the end.




```
variables
output x:ℝ:= x0; y:ℝ:= y0;                                   2
  θ:ℝ:= θ0; v:ℝ:= v0
input a, φ: ℝ                                                4

trajectories                                                 6
evolve d(x) = v cos(θ)
  d(y) = v sin(θ)                                            8
  if |u.φ| ≤ φmax
    then d(θ) = v/L tan(φ)                                   10
    else d(θ) = v/L tan(φ/|φ| φmax) fi
  if v > 0 ∨ a ≥ 0                                           12
    then d(v) = a
    else d(v) = 0 fi                                         14
```

(c)

Fig. 4. (a) Planner-Controller system. (b) Deviation & disorientation. (c) Vehicle.

with respect to the positive direction of the $x$ axis, and (c) vehicle's velocity $v$. These variables evolve according to the differential equations of lines 7–14. Two aspects of this Vehicle model are noteworthy:

(i) In determining the orientation of the vehicle, if the input steering angle $\phi$ is greater than the maximum limit $\phi_{max}$ then the maximum steering in the correct direction is applied.

(ii) The acceleration can be negative only if the velocity is positive, and therefore the vehicle cannot move backwards.

This vehicle model requires bounds on minimum and maximum acceleration, however, the controller ensures that the input acceleration is always within such a bound.

### 4.2 Controller

Figure 5 shows the SHIOA specification of the Controller automaton that reads the state of the Vehicle periodically and issues acceleration and steering outputs to achieve the aforementioned goals.

Controller is parameterized by: (a) the sampling period $\Delta \in \mathbb{R}_+$ , (b) the target speed $v_T \in \mathbb{R}_{\geq 0}$, (c) proportional control gains $k_1, k_2 > 0$, (d) a constant $\delta > 0$ relating the maximum steering angle and the speed, (e) maximum and braking accelerations $a_{max} > 0$ and $a_{brake} < 0$. Restricting the maximum steering angle instead of the maximum steering rate is a simplifying but conservative assumption.

Given a constant relating the maximum steering rate and the speed, there exists $\delta$ as defined above that guarantees that the maximum steering rate requirement is satisfied.

A *path* is an infinite sequence of points $p_1, p_2, \ldots$ where $p_i \in \mathbb{R}^2$, for each $i$. The main state variables of Controller are the following:

(a) *brake* and *new_path* are command variables that store the information received through the most recent brake (*On* or *Off*) and plan (a path) actions.

(b) *path* is the current path being followed by Controller,

(c) *seg* is the index of the last waypoint visited in the current *path*. That is, $seg+1$ is the index of the current waypoint. The straight line joining $path[seg]$ and $path[seg + 1]$ is called the *current segment*.

(d) *deviation* $e_1$ is the signed perpendicular distance from the current position of the vehicle to the current segment (see, Figure 4(b)).

(e) *disorientation* $e_2$ is the difference between the current orientation of the vehicle ($\theta$) and the angle of the current segment.

(f) *waypoint-distance* $d$ is the signed distance of the vehicle to the current waypoint measured parallel to the current segment.

**signature**
  **input** plan($p$:Seq[$\mathbb{R}^]$); brake($b : On, Off$)
  **internal** main

**variables**
  **input** $x, y, \theta, v$: $\mathbb{R}$
  **output** $a, \phi$: $\mathbb{R} :=$ (0, 0)
  **internal** *brake*: {*On*, *Off*} := *Off*
    *path*: Seq[$\mathbb{R}^2$] := *arbitrary*
    *new_path*: Seq[$\mathbb{R}^2$] := *path*
    *seg*: $\mathbb{N} := 1$
    $e_1, e_2, d : \mathbb{R} := [e_{1,0}, e_{2,0}, d_0]$
    *now*: $\mathbb{R} := 0$; *next*:$\mathbb{R}_{\geq 0} := 0$

**transitions**
  **input** plan($p$)
  **eff** *new_path* := $p$

  **input** brake($b$)
  **eff** *brake* := $b$

  **internal** main
  **pre** $now = next$
  **eff** $next := now + \Delta$
  **if** $path \neq new\_path \vee d \leq 0$ **then**
    **if** $path \neq new\_path$
      **then** $seg := 1$; $path := new\_path$
    **elseif** $d \leq 0$
      **then** $seg := seg + 1$ **fi**

    **let** $\vec{p} = \begin{bmatrix} path[seg+1].x - path[seg].x \\ path[seg+1].y - path[seg].y \end{bmatrix}$
    $\vec{q} = \begin{bmatrix} path[seg+1].y - path[seg].y \\ -(path[seg+1].x - path[seg].x) \end{bmatrix}$
    $\vec{r} = \begin{bmatrix} path[seg+1].x - x \\ path[seg+1].y - y) \end{bmatrix}$
    $e_1 := \frac{1}{\|\vec{q}\|}\vec{q} \cdot \vec{r}$
    $e_2 := \theta - \angle\vec{p}$
    $d := \frac{1}{\|\vec{p}\|}\vec{p} \cdot \vec{r}$
  **fi**

  **let** $\phi_d = -k_1\ e_1 - k_2\ e_2$
  $\phi = \frac{\phi_d}{|\phi_d|} \min(\delta \times v, |\phi_d|)$

  **if** $brake = On$ **then** $a := a_{brake}$
  **elseif** $brake = Off \wedge v < v_T$
    **then** $a := a_{max}$
  **else** $a := 0$ **fi**

**trajectories**
  **evolve** $d(now) = 1$
    $d(e_1) = v \sin(e_2)$
    $d(e_2) = \frac{v}{L} \tan(\phi)$
    $d(d) = -v \cos(e_2)$
  **stop when** $now = next$

Fig. 5. Controller with parameters $v_T, k_1, k_2 \in \mathbb{R}_{\geq 0}$, $\delta, \Delta \in \mathbb{R}_+$ and $a_{brake} < 0$.

The brake($b$) action is an externally controlled input action that informs the Controller about the application of an external brake ($b = On$) or the removal of the brake ($b = Off$). When brake($b$) occurs, $b$ is recorded in the command variable

*brake*. The plan($p$) action is controlled by the external Planner (not presented in this paper) and it informs the Controller about a newly planned path $p$. When this action occurs, the path $p$ is recorded in the variable *new_path*. The main action occurs once every $\Delta$ time starting from time 0. This action updates the values of the variables $e_1, e_2, d, path, seg, a$ and $\phi$ as follows:

A. If *new_path* (obtained from the planner) is different from *path* then *seg* is set to 1 and *path* is set to *new_path* (line 27).
B. If *new_path* is the same as *path* and the waypoint-distance $d$ is less than or equal to 0, then *seg* is set to $seg + 1$ (line 29).
C. For both of the above cases several temporary variables are computed that are in turn used to update $e_1, e_2, d$ as specified in Lines 33-35; otherwise these variables remain unchanged.
D. The steering output to the vehicle $\phi$ is computed using a proportional control law and it is restricted to be at most $\delta$ times the velocity of the vehicle for the mechanical protection of the steering. That is, the magnitude of the steering output $\phi$ is set to the minimum of $|-k_1e_1 - k_2e_2|$ and $v \times \delta$ (line 39).
E. The acceleration output $a$ is computed using bang bang control law. If *brake* is *On* then $a$ is set to the braking deceleration $a_{brake}$; otherwise, it executes $a_{max}$ until the vehicle reaches the target speed, at which point $a$ is set to 0.

Along a trajectory, the evolution of the variables are specified by the differential equations on lines 48-50. These differential equations are derived from the update rules described above and the differential equations governing the evolution of $x, y, \theta$ and $v$.

### 4.3 Complete System

Let $\mathcal{A}$ be the composition of the Controller and the Vehicle automata. The continuous state of $\mathcal{A}$ is defined by the valuations of $x, y, \theta, v, e_1, e_2$, and $d$ of Vehicle and Controller. For convenience, we define a single derived variable $s$ of type $\mathcal{X} = \mathbb{R}^7$ encapsulating all these variables. The discrete state of $\mathcal{A}$ is defined by the valuations of $brake, path$ and $seg$ of Controller. A derived variable *loc* of type $\mathcal{L} = \texttt{Tuple}[\{On, Off\}, \texttt{Seq}[\mathbb{R}^2], \mathbb{N}]$ is defined encapsulating all these variables. It can be checked easily that the composed automaton $\mathcal{A}$ is a PCHA. Appendix A describes the variables, actions, state transition functions of the corresponding PCHA.

## 5. ANALYSIS OF THE SYSTEM

Overview. The informally stated goals of the system translate to the following subgoals:

A. (*safety)* At all reachable states of $\mathcal{A}$, the deviation ($e_1$) of the vehicle is upper-bounded by $e_{max}$, where $e_{max}$ is determined in terms of system parameters.
B. (*segment progress)* There exist certain threshold values of deviation, disorientation, and waypoint-distance such that from any state $\mathbf{x}$ with greater deviation, disorientation and waypoint-distance, the vehicle reduces its deviation and disorientation with respect to the current segment, while making progress towards its current waypoint.

C. (*waypoint progress)* The vehicle reaches successive waypoints.

First, in Sections 5.1 and 5.2, we define a family $\{\mathcal{I}_k\}_{k\in\mathbb{N}}$ of subsets of $Q_{\mathcal{A}}$ and using Lemma 3.2 and Lemma 3.3, we conclude that they are invariant with respect to the control-free execution fragments of $\mathcal{A}$. From the specification of main action, we see that the continuous state changes only occur if $path \neq new\_path$ or waypoint-distance $d \leq 0$. Hence, using Theorem 3.4, we conclude that any execution fragment starting in $\mathcal{I}_k$ remains within $\mathcal{I}_k$, provided that *path* and current segment do not change.

In Section 5.3, we establish the segment progress property (B) by showing that starting from $\mathcal{I}_k$, $\mathcal{I}_{k+1}$ is reached in a finite amount of time and for $k$ smaller than the threshold value $k^*$, $\mathcal{I}_{k+1}$ is strictly contained in $\mathcal{I}_k$. Finally, in Section 5.4, we prove an invariance of $\mathcal{I}_0$ and derive geometric properties of planner paths that can be followed by $\mathcal{A}$ safely. These geometric properties specify the minimum length of a path segment and the relationship between the segment length and the maximum difference between consecutive segment orientations and are derived from the segment progress property. An invariance of $\mathcal{I}_0$ provides a proof certificate that $\mathcal{A}$ satisfies the safety property (A) and the waypoint progress property (C).

### 5.1 Family of Invariants

We define, for each $k \in \mathbb{N}$, the set $\mathcal{I}_k$ that bounds the deviation of the vehicle $e_1$ to be within $[-\epsilon_k, \epsilon_k]$. This bound on deviation alone, of course, does not give us an inductive invariant. If the deviation is $\epsilon_k$ and the vehicle is highly disoriented, then it would violate $\mathcal{I}_k$. Thus, $\mathcal{I}_k$ also bounds the disorientation such that the steering angle computed based on the proportional control law is within $[-\phi_k, \phi_k]$. To prevent the vehicle from not being able to turn at low speed and to guarantee that the execution speed of the controller is fast enough with respect to the speed of the vehicle, $\mathcal{I}_k$ also bounds the speed of the vehicle. $\mathcal{I}_k$ is defined in terms of $\epsilon_k, \phi_k \geq 0$ as

$$\mathcal{I}_k \triangleq \{\mathbf{x} \in Q \mid \forall i \in \{1, \ldots 6\}, F_{k,i}(\mathbf{x}.s) \geq 0\} \tag{11}$$

where $F_{k,1}, \ldots, F_{k,6} : \mathbb{R}^7 \to \mathbb{R}$ are defined as follows:

$$F_{k,1}(s) = \epsilon_k - s.e_1; \qquad F_{k,2}(s) = \epsilon_k + s.e_1; \tag{12}$$

$$F_{k,3}(s) = \phi_k + k_1 s.e_1 + k_2 s.e_2; \qquad F_{k,4}(s) = \phi_k - k_1 s.e_1 - k_2 s.e_2; \tag{13}$$

$$F_{k,5}(s) = v_{max} - s.v; \qquad F_{k,6}(s) = \delta s.v - \phi_b. \tag{14}$$

Here $v_{max} = v_T + \Delta a_{max}$ and $\phi_b > 0$ is an arbitrary constant. As we shall see shortly, the choice of $\phi_b$ affects the minimum speed of the vehicle and also the requirements of a brake action. We examine a state $\mathbf{x} \in \mathcal{I}_k$, that is, $F_{k,i}(\mathbf{x}.s) \geq 0$ for any $i \in \{1, \ldots 6\}$. $F_{k,1}(s), F_{k,2}(s) \geq 0$ means $s.e_1 \in [-\epsilon_k, \epsilon_k]$. $F_{k,3}(s), F_{k,4}(s) \geq 0$ means that the steering angle computed based on the proportional control law is in the range $[-\phi_k, \phi_k]$. Further, if $\phi_k \leq \phi_{max}$, then the computed steering satisfies the physical constraint of the vehicle. If, in addition, we have $\phi_b \geq \phi_k$ and $F_{k,6}(s) \geq 0$, then the vehicle actually executes the computed steering command. $F_{k,5}(s) \geq 0$ means that the speed of the vehicle is at most $v_{max}$. The sets $\mathcal{I}_k$, projected onto the $(e_1, e_2)$ plane, for different values of the parameters $\epsilon_k$ and $\phi_k$ are shown in Figure 6.

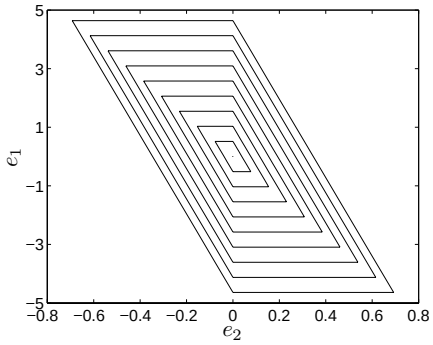



Fig. 6. The set $\mathcal{I}_k$ for different values of $\epsilon_k$ and $\phi_k$, projected onto the $e_1, e_2$ plane.

For each $k \in \mathbb{N}$, we define

$$\theta_{k,1} = \frac{k_1}{k_2}\epsilon_k - \frac{1}{k_2}\phi_k \tag{15}$$

$$\theta_{k,2} = \frac{k_1}{k_2}\epsilon_k + \frac{1}{k_2}\phi_k \tag{16}$$

That is, $\theta_{k,1}$ and $\theta_{k,2}$ are the values of $e_2$ at which the proportional control law yields the steering angle of $\phi_k$ and $-\phi_k$ respectively, given that the value of $e_1$ is $-\epsilon_k$. From the above definitions, we make the following observations about the boundary of the $\mathcal{I}_k$ sets: for any $k \in \mathbb{N}$ and $\mathbf{x} \in \mathcal{I}_k$,

(a) $\mathbf{x}.e_2 \in [-\theta_{k,2}, \theta_{k,2}]$,
(b) $F_{k,1}(\mathbf{x}.s) = 0$ implies $\mathbf{x}.e_2 \in [-\theta_{k,2}, -\theta_{k,1}]$,
(c) $F_{k,2}(\mathbf{x}.s) = 0$ implies $\mathbf{x}.e_2 \in [\theta_{k,1}, \theta_{k,2}]$,
(d) $F_{k,3}(\mathbf{x}.s) = 0$ implies $\mathbf{x}.e_2 \in [-\theta_{k,2}, \theta_{k,1}]$, and
(e) $F_{k,4}(\mathbf{x}.s) = 0$ implies $\mathbf{x}.e_2 \in [-\theta_{k,1}, \theta_{k,2}]$.

We assume that $\phi_b$ and all the $\epsilon_k's$ and $\phi_k$'s satisfy the following assumptions that are derived from physical and design constraints on the controller. The region in the $\phi_k$,$\epsilon_k$ plane that satisfies Assumption 5.1 is shown Figure 7.

***Assumption* 5.1.** (Vehicle and controller design)

(a) $\phi_k \leq \phi_b \leq \phi_{max}$ and $\phi_k < \frac{\pi}{2}$
(b) $0 \leq \theta_{k,1} \leq \theta_{k,2} < \frac{\pi}{2}$
(c) $L \cot \phi_k \sin \theta_{k,2} < \frac{k_2}{k_1}$
(d) $\Delta \leq \frac{c}{b}$ where $c = \frac{1}{\sqrt{k_1^2+k_2^2}}(\phi_k - \tilde{\phi})$, $b = v_{max}\sqrt{\sin^2\theta_{k,2} + \frac{1}{L^2}\tan^2(\tilde{\phi})}$ and $\tilde{\phi} = \cot^{-1}\left(\frac{k_2}{k_1 L \sin\theta_{k,2}}\right)$. [6]
(e) $\frac{\tan\phi_k}{2L} v_{max} \Delta \leq \frac{\pi}{2}$

[6] Using Assumption 5.1(c), it can be shown that $\tilde{\phi} < \phi_k$ so $\frac{c}{b} > 0$.

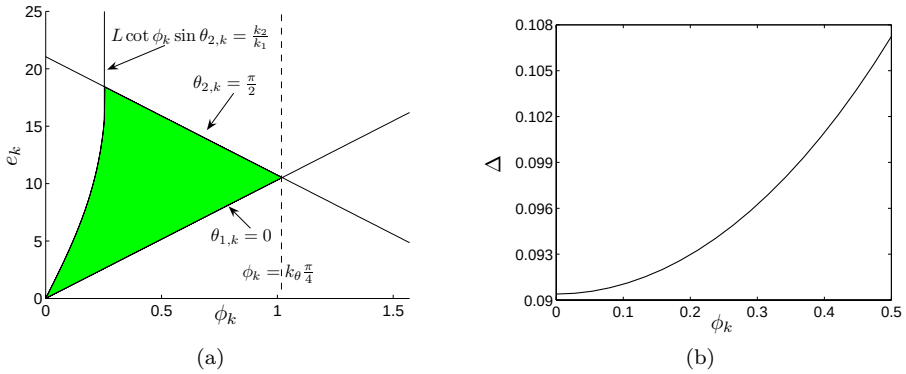



Fig. 7. (a) The set of ($\epsilon_k$, $\phi_k$) which satisfies Assumptions 5.1 (c) and (d) and are represented by the green region. (b) The relationship between the maximum bound on $\Delta$ and $\phi_k$ for $\epsilon_k = \frac{1}{k_1}\phi_k$.

If the vehicle is forced to slow down too much at the boundary of an $\mathcal{I}_k$ by the brakes, then it may not be able to turn enough to remain inside $\mathcal{I}_k$. Thus, in verifying the above properties we need to restrict our attention to executions in certain *good* brake controller in which brake inputs do not occur at low speeds and are not too persistant. This is formalized by the next definition.

***Definition* 5.2.** A brake controller is *good* if its composition with controller gives rise to controller executions that satisfy: if a brake($On$) action occurs at time $t$ then (a) $\alpha(t).v > \frac{\phi_b}{\delta} + \Delta|a_{brake}|$, and (b) brake($Off$) must occur within time $t + \frac{1}{|a_{brake}|}(\alpha(t).v - \frac{\phi_b}{\delta} - \Delta|a_{brake}|)$.

We assume that the brake controller satisfies the above assumption and for the remainder of this section, we only consider executions in *good* brake controller. A state $\mathbf{x} \in Q_{\mathcal{A}}$ is reachable if there exists an execution in *good* brake controller $\alpha$ with $\alpha.\mathsf{lstate} = \mathbf{x}$.

## 5.2 Invariance Property

We fix a $k \in \mathbb{N}$ for the remainder of the section and denote $\mathcal{I}_k, F_{k,i}$ as $\mathcal{I}$ and $F_i$, respectively, for $i \in \{1, \ldots, 6\}$. As in Lemma 3.2, we define $I = \{s \in \mathcal{X} \mid F_i(s) \geq 0\}$ and for each $i \in \{1, \ldots, 6\}$, $\partial I_i = \{s \in \mathcal{X} \mid F_i(s) = 0\}$ and let the functions $f_1, f_2, \ldots, f_7 : \mathbb{R}^7 \times \mathbb{R}^2 \to \mathbb{R}$ as defined in Appendix A describe the evolution of $x$, $y$, $\theta$, $v$, $e_1$, $e_2$ and $d$, respectively. We prove that $I$ satisfies the control-free invariance condition of Lemma 3.1 by applying Lemma 3.2.

First, we define the sets $C_1, \ldots, C_6$ and show that all the assumptions in Lemma 3.2 are satisfied. The proof does not involve solving differential equations but requires algebraic simplification of the expressions defining the vector fields and the bound-

aries $\{\partial I_i\}_{i\in\{1,\ldots 6\}}$ of the invariant set.

$$C_1 = C_2 = \emptyset \tag{17}$$

$$C_3 = \{s \in I \mid -k_1 s.e_1 - k_2 s.e_2 \leq 0 \vee L\cot(-k_1 s.e_1 - k_2 s.e_2)\sin\theta_{k,2} \geq \frac{k_2}{k_1}\} \tag{18}$$

$$C_4 = \{s \in I \mid -k_1 s.e_1 - k_2 s.e_2 \geq 0 \vee L\cot(k_1 s.e_1 + k_2 s.e_2)\sin\theta_{k,2} \geq \frac{k_2}{k_1}\} \tag{19}$$

$$C_5 = \{s \in I \mid s.v \leq v_T\} \tag{20}$$

$$C_6 = \{s \in I \mid s.v \geq \frac{\phi_b}{\delta} + \Delta|a_{brake}|\} \tag{21}$$

From the definition of a *good* brake controller (Definition 5.2), we show that when the value of the variable *brake* is *On*, the speed of the vehicle is at least $\frac{\phi_b}{\delta} + \Delta|a_{brake}|$.

**Lemma 5.3.** *At any reachable state* $\mathbf{x}$ *of* $\mathcal{A}$*, if* $\mathbf{x}.brake = On$ *then* $\mathbf{x}.v \geq \frac{\phi_b}{\delta} + \Delta|a_{brake}|$.

Proof. Consider an arbitrary execution fragment, $\alpha = \tau_0 a_1 \tau_1 a_2 \ldots$ and an arbitrary $i \in \mathbb{N}$ such that $(\tau_i \downarrow brake)(0) = On$. Since the initial value of the variable *brake* is *Off*, there must exist $j \leq i$ such that $a_j$ is a $\mathsf{brake}(On)$ action and for any natural number $m \in [j, i]$, $a_m$ is not a $\mathsf{brake}(Off)$ action. Let $(\tau_{j-1}.\mathsf{lstate}) \lceil v = v_b$. Since $a_j$ is a $\mathsf{brake}(On)$ action which does not affect $v$, we get $(\tau_j.\mathsf{fstate}) \lceil v = v_b$. From Definition 5.2, $v_b > \frac{\phi_b}{\delta} + \Delta|a_{brake}|$ and there must exist $k > i$ such that $a_k$ is a $\mathsf{brake}(Off)$ action and $\sum_{m=j}^{k-1} \tau_m.\mathsf{ltime} \leq \frac{1}{|a_{brake}|}(v_b - \frac{\phi_b}{\delta} - \Delta|a_{brake}|)$. So for any $t \in dom(\tau_i)$, we get

$$\begin{aligned}(\tau_i \downarrow v)(t) &\geq v_b + \min_{s,s_0\in\mathcal{X},l\in\mathcal{L}} f_4(s, g(l, s_0))(t + \sum_{m=j}^{i-1} \tau_m.\mathsf{ltime}) \\ &\geq v_b + a_{brake}(\sum_{m=j}^{k-1} \tau_m.\mathsf{ltime}) = \frac{\phi_b}{\delta} + \Delta|a_{brake}|.\end{aligned}$$

□

The next lemma shows that the subtangential, bounded distance and bounded speed conditions (of Lemma 3.2) are satisfied with the the sets $\{C_j\}_{j\in\{1,\ldots,6\}}$ defined in (17)-(21). The proof applies Lemma 3.3. The knowledge about the reachable state $\mathbf{x}$ of $\mathcal{A}$ with $\mathbf{x}.brake = On$, provided in Lemma 5.3, is needed to prove the subtangential condition for $j = 6$.

**Lemma 5.4.** *For each* $l \in \mathcal{L}$ *and* $j \in \{1, \ldots, 6\}$*, the subtangential, bounded distance, and bounded speed conditions (of Lemma 3.2) are satisfied.*

Proof. Since $C_1, C_2 = \emptyset$, we see that the bounded distance and bounded speed conditions are automatically satisfied for $j = 1, 2$ with any arbitrary large $c_j$ and arbitrary small $b_j$. Now, consider an arbitrary $s_0 \in I$ and $s \in \partial I_1$. By definition, $F_1(s) = 0$. From the definition of $\theta_{k,1}$ and $\theta_{k,2}$ and Assumption 5.1(b), $s.e_2 \in [-\theta_{k,2}, -\theta_{k,1}] \subset (-\frac{\pi}{2}, 0]$. In addition, since $s \in I$, $F_6(s) = \delta s.v - \phi_b \geq 0$ and since

$\delta > 0$ and $\phi_b \geq 0$, $s.v \geq 0$. Thus,

$$\frac{\partial F_1}{\partial s}(s) \cdot f(s, g(l, s_0)) = -\frac{de_1}{dt} = -s.v \sin(s.e_2) \geq 0$$

For $j = 2$, the subtangential condition can be proved in a similar way.

To prove the bounded distance and the bounded speed conditions for $j = 3, \ldots, 6$, we apply Lemma 3.3. Let $\mathcal{U}_I = \{g(l, s) \mid l \in \mathcal{L},\ s \in I\}$. From the definition of $I$, we get that for any $s_0 \in I$, $-k_1 s_0.e_1 - k_2 s_0.e_2 \in [-\phi_k, \phi_k] \subset (-\frac{\pi}{2}, \frac{\pi}{2})$. Therefore, $f$ is continuous in $I \times \mathcal{U}_I$.

In addition, it can be easily checked that the projection of $I$ onto the $(e_1, e_2, v)$ space is compact and for any $j \in \{3, \ldots, 6\}$, $C_j$ is closed. Since the only variables involved in proving the control-free invariance condition of Lemma 3.1 are $e_1$, $e_2$ and $v$ whose evolution along a trajectory can be described without other variables, from the proof of Lemma 3.2 and Lemma 3.3, we see that the requirement that $I$ is compact can be relaxed to the requirement the projection of $I$ onto the $(e_1, e_2, v)$ space is compact. Hence, from Lemma 3.3, to prove that conditions (a)–(c) of Lemma 3.2 hold, we only need to show that for any $l \in \mathcal{L}$, the following conditions are satisfied for each $j \in \{3, \ldots, 6\}$:

(1) $C_j \cap \partial I_j = \emptyset$

(2) For any $s_0 \in I \setminus C_j$ and $s \in \partial I_j$, $\frac{\partial F_j}{\partial s} \cdot f(s, g(l, s_0)) \geq 0$

Consider an arbitrary $s \in \partial I_3$. From the definition of $I_3$, $-k_1 s.e_1 - k_2 s.e_2 = \phi_k > 0$. So from Assumption 5.1(c), $L \cot(-k_1 s.e_1 - k_2 s.e_2) \sin\theta_{k,2} < \frac{k_2}{k_1}$. Therefore, $C_3 \cap \partial I_3 = \emptyset$. Pick an arbitrary $s_0 \in I \setminus C_3$. From the definition of $I$ and $C_3$, $0 < -k_1 s_0.e_1 - k_2 s_0.e_2 \leq \phi_k$ and $L \cot(-k_1 s_0.e_1 - k_2 s_0.e_2) \sin\theta_{k,2} < \frac{k_2}{k_1}$. Combining this with Assumption 5.1(a), we get $0 < -k_1 s_0.e_1 - k_2 s_0.e_2 \leq \frac{\pi}{2}$ and $|-k_1 s_0.e_1 - k_2 s_0.e_2| \leq \phi_{max}$. In addition, since $s_0 \in I$, $F_6(s_0) \geq 0$ and so $\delta s_0.v \geq \phi_b \geq \phi_k \geq |-k_1 s_0.e_1 - k_2 s_0.e_2|$, and since $s \in I$, $s.v \geq 0$. Therefore, we can conclude that

$$\frac{ds.e_2}{dt} = \frac{s.v}{L} \tan(-k_1 s_0.e_1 - k_2 s_0.e_2) \geq 0$$

and from Assumption 5.1(b), $s.e_2 \in [-\theta_{k,2}, \theta_{k,1}] \subset (-\frac{\pi}{2}, 0]$. So we get

$$\begin{aligned} \frac{ds.e_1}{ds.e_2} &= L \cot(-k_1 s_0.e_1 - k_2 s_0.e_2) \sin(s.e_2) \\ &\geq -L \cot(-k_1 s_0.e_1 - k_2 s_0.e_2) \sin\theta_{k,2} \\ &> -\frac{k_2}{k_1}. \end{aligned}$$

Thus,

$$\frac{\partial F_3}{\partial s} \cdot f(s, g(l, s_0)) = k_2 \frac{ds.e_2}{dt} + k_1 \frac{ds.e_1}{dt} = \frac{ds.e_2}{dt} \left( k_2 + k_1 \frac{ds.e_1}{ds.e_2} \right) \geq 0.$$

This completes the proof for $j = 3$.

For $j = 4$, we can follow the previous proof to show that $C_4 \cap \partial I_4 = \emptyset$, $\frac{ds.e_2}{dt} \leq 0$ and $\frac{ds.e_1}{ds.e_2} > -\frac{k_2}{k_1}$, and so

$$\forall s_0 \in I \setminus C_4, \frac{\partial F_4}{\partial s} \cdot f(s, g(l, s_0)) \geq 0.$$

Next, consider an arbitrary $s \in \partial I_5$. From the definition of $\partial I_5$, $s.v = v_{max}$. Since $a_{max}, \Delta > 0$, $v_{max} = v_T + \Delta a_{max} > v_T$. Therefore, $C_5 \cap \partial I_5 = \emptyset$. Pick an arbitrary $s_0 \in I \setminus C_5$. From the definition of $I$ and $C_5$, $v_T < s_0.v \leq v_{max}$. Therefore, we can conclude that

$$\frac{\partial F_5}{\partial s} \cdot f(s, g(l, s_0)) = \begin{cases} -a_{brake} \\ 0 \end{cases} \geq 0.$$

This completes the proof for $j = 5$.

Finally, consider an arbitrary $s \in \partial I_6$. From the definition of $\partial I_6$, $s.v = \frac{\phi_b}{\delta}$. Since $\Delta, |a_{brake}| > 0$, $\frac{\phi_b}{\delta} < \frac{\phi_b}{\delta} + \Delta|a_{brake}|$. Therefore, $C_6 \cap \partial I_6 = \emptyset$. Consider an arbitrary $s_0 \in I \setminus C_6$. From Lemma 5.3 and the definition of $f_4$, we see that $f_4(s, g(l, s_0)) = a_{brake}$ only if $s_0.v \geq \frac{\phi_b}{\delta} + \Delta|a_{brake}|$. But since $s_0 \in I \setminus C_6$, from the definition of $I$ and $C_6$, $s_0.v < \frac{\phi_b}{\delta} + \Delta|a_{brake}|$. Therefore, $f_4(s, g(l, s_0))$ is either 0 or $a_{max}$ and so we can conclude that

$$\frac{\partial F_6}{\partial s} \cdot f(s, g(l, s_0)) = f_4(s, g(l, s_0)) \geq 0.$$

□

From the definition of each $C_j$, we can derive the lower bound $c_j$ on the distance from $C_j$ to $\partial I_j$ and the upper bound $b_j$ on the length of the vector field $f$ where the control variable $u$ is evaluated when the continuous state $s \in C_j$. Using these bounds, we prove the sampling rate condition.

**Lemma 5.5.** *For each $l \in \mathcal{L}$, the sampling rate condition (of Lemma 3.2) is satisfied.*

PROOF. For each $j \in \{1, \ldots, 6\}$, we want to find $c_j$ and $b_j$ which satisfy condition (b) and (c) of Lemma 3.2. First, we note that for $j = 1, 2$, $C_j = \emptyset$, so $c_j$ can be arbitrary large and $b_j$ can be arbitrary small and therefore any $\Delta \in \mathbb{R}_+$ satisfies the sampling rate condition of Lemma 3.2. For $j = 5, 6$, it can be easily shown that $c_5 = \Delta a_{max}$, $b_5 = a_{max}$, $c_6 = \Delta|a_{brake}|$ and $b_6 = |a_{brake}|$; thus, $\frac{c_j}{b_j} = \Delta$. That is, $\Delta$ can be an arbitrary large number if we only consider $j = 1, 2, 5, 6$. So we only have to consider $j = 3, 4$. From Assumption 5.1(c), there exists

$$\tilde{\phi} = \cot^{-1}\left(\frac{k_2}{k_1 L \sin\theta_{k,2}}\right) < \phi_k.$$

Using symmetry, we get that for $j = 3$ and $j = 4$, the shortest distance between $\mathcal{U}_j$ and $\partial I_j$ is then given by

$$c_j = \min_{s \in \partial I_j, s_0 \in \mathcal{U}_j} \|s - s_0\| = \frac{1}{\sqrt{k_1^2 + k_2^2}}(\phi_k - \tilde{\phi}).$$

Since $\forall s \in I, s.e_2 \in [-\theta_{k,2}, \theta_{k,2}] \subset (-\frac{\pi}{2}, \frac{\pi}{2})$, we have

$$\begin{aligned} b_j &= \max_{s \in I, s_0 \in \mathcal{U}_j} \|f(s, g(l, s_0))\| \\ &\leq v_{max}\sqrt{\sin^2\theta_{k,2} + \frac{1}{L^2}\tan^2(\tilde{\phi})}. \end{aligned}$$

From Assumption 5.1(d), we see that $\Delta \leq \min_{j\in\{1,\ldots,6\}} \frac{c_j}{b_j}$. □

Thus, all assumptions in the hypothesis of Lemma 3.2 are satisfied; from Theorem 3.4 we obtain that execution fragments in *good* brake controller of $\mathcal{A}$ preserve invariance of $\mathcal{I}$, provided that the path and current segment do not change over the fragment.

**Theorem 5.6.** *For any* plan*-free execution fragment $\beta$ starting at a state $\mathbf{x} \in \mathcal{I}$ and ending at $\mathbf{x}' \in Q_{\mathcal{A}}$, if $\mathbf{x}.path = \mathbf{x}.new\_path$ and $\mathbf{x}.seg = \mathbf{x}'.seg$, then $\mathbf{x}' \in \mathcal{I}$.*

Proof. From Lemmas 5.4–5.5, we see that all the conditions in Lemma 3.2 are satisfied. Thus, we can conclude that the control-free invariance condition of Lemma 3.1 is satisfied. In addition, from the specification of main action, we see that a discrete transition in the continuous state $s$ only occurs when $path \neq new\_path$ (i.e. a new path is received) or $s.d \leq 0$ (i.e. the vehicle has reached the end of the current segment). Hence, if a closed execution $\beta$ does not contain a plan action, $\beta.\mathsf{fstate} \lceil path = \beta.\mathsf{fstate} \lceil new\_path$ and $\beta.\mathsf{lstate} \lceil seg = \beta.\mathsf{fstate} \lceil seg$, then a discrete transition in the continuous state $s$ does not occur in $\beta$. Applying Theorem 3.4, we get the desired result. □

### 5.3 Segment Progress

In this section, we establish the segment progress property, i.e., there exist certain threshold values of deviation, disorientation, and waypoint-distance such that from any state $\mathbf{x}$ with greater deviation, disorientation and waypoint-distance, the vehicle reduces its deviation and disorientation with respect to the current segment, while making progress towards its current waypoint. First, we prove the progress property over a pasted trajectory $\tau$ between any two main actions. That is, suppose right after an occurrence of a main action, $\mathbf{x} \in \mathcal{I}_k$ for some $k \in \mathbb{N}$. Then, right before an occurrance of the next main action, $\mathbf{x} \in \mathcal{I}_{k+1}$ where $\mathcal{I}_{k+1} \subseteq \mathcal{I}_k$ and if $k$ is less than some threshold $k^*$, then $\mathcal{I}_{k+1}$ is strictly contained in $\mathcal{I}_k$.

Next, in Lemma 5.9, we compute the bound $d^*$ on the maximum change in the value of the waypoint distance $d$ over $\tau$. Given the progress property over $\tau$ and the bound $d^*$, we can then establish the segment progress property (B) defined at the beginning of Section 5. That is, starting from a state $\mathbf{x}$ and ending at $\mathbf{x}'$, if $\mathbf{x} \in \mathcal{I}_k$, then $\mathbf{x}' \in \mathcal{I}_{k+n}$ where an integer $n \geq 0$ depends on $\mathbf{x}.d - \mathbf{x}'.d$ and the system parameters, provided that path and current segment do not change. Furthermore, if $\mathbf{x}.d - \mathbf{x}'.d$ is large enough, then $n$ is strictly positive.

First, we solve the differential equation which describes the evolution of $e_1$ and $e_2$ along $\tau$. From periodicity of main actions we see that $dom(\tau) = [0, \Delta]$. Define the functions $e_1, e_2, v, v_{avg} : dom(\tau) \to \mathbb{R}$ as follows: $e_1(t) = (\tau \downarrow e_1)(t)$, $e_2(t) = (\tau \downarrow e_2)(t)$, $v(t) = (\tau \downarrow v)(t)$ and $v_{avg}(t) = \frac{1}{t}\int_0^t v(t')dt'$. From the state models of the Vehicle and the Controller specified in Figure 4 and Figure 5, since $\phi$ and $a$ are constant along $\tau$, the solution to the differential equations can be solved analytically and are given by

$$
\begin{aligned}
e_1(t) &= \begin{cases} e_1(0) + L\cot\phi\cos e_2(0) - L\cot\phi\cos e_2(t) & \text{if } \phi \neq 0 \\ e_1(0) + v_{avg}(t)t\sin e_2(0) & \text{otherwise} \end{cases} \\
e_2(t) &= e_2(0) + \tfrac{\tan\phi}{L} v_{avg}(t)t
\end{aligned}
\tag{22}
$$

where $\phi = \tau.\mathsf{fstate} \lceil \phi$ and $a = \tau.\mathsf{fstate} \lceil a$.

The following lemma provides a bound on the change in $e_1$ over $\tau$ and on the change in $\phi$ between two consecutive main actions assuming that a discrete transition in the continuous state $s$ does not occur.

**Lemma 5.7.** *Suppose $\tau.\mathsf{fstate} \in \mathcal{I}_k$ for some $k \in \mathbb{N}$. Then, $|e_1(0) - e_1(\Delta)| \leq \Delta_e$ and $|(k_1 e_1(0) + k_2 e_2(0)) - (k_1 e_1(\Delta) + k_2 e_2(\Delta))| \leq \Delta_\phi$ where $\Delta_e = v_{max}\Delta$ and $\Delta_\phi = v_{max}\Delta\left(k_1 + k_2\frac{\tan\phi_k}{L}\right)$.*

Proof. From (22), we see that $|e_1(\Delta) - e_1(0)| \leq v_{max}\Delta$ and $|e_2(\Delta) - e_1(0)| \leq \frac{\tan\phi_k}{L} v_{max}\Delta$. So

$$\begin{aligned}|(k_1 e_1(0) + k_2 e_2(0)) - (k_1 e_1(\Delta) + k_2 e_2(\Delta))| &\leq k_1|e_1(\Delta) - e_1(0)| + k_2|e_2(\Delta) - e_2(0)| \\ &\leq k_1 v_{max}\Delta + k_2\frac{\tan\phi_k}{L} v_{max}\Delta.\end{aligned}$$

□

The next lemma proves the desired progress property over $\tau$.

**Lemma 5.8.** *Suppose $\tau.\mathsf{fstate} \in \mathcal{I}_k$ for some $k \in \mathbb{N}$. Then $\tau.\mathsf{lstate} \in \mathcal{I}_{k+1}$ whose parameters $\epsilon_{k+1}$ and $\phi_{k+1}$ are given by*

$$\epsilon_{k+1} = \epsilon_k - a_k \tag{23}$$

$$\phi_{k+1} = \phi_k - b_k \tag{24}$$

*where $a_k, b_k \geq 0$ and are given by*

$$a_k = \epsilon_k - \max\left(\epsilon'_{k+1}, \frac{1}{k_1}\phi'_{k+1}\right) \tag{25}$$

$$b_k = \phi_k - \max(\phi'_{k+1}, \varphi) \tag{26}$$

$$\epsilon'_{k+1} = \begin{cases} \max\left(\epsilon_k - \xi_k, \epsilon_k^*\right) & \text{if } \epsilon_k > \epsilon_k^* \\ \epsilon_k & \text{otherwise} \end{cases} \tag{27}$$

$$\phi'_{k+1} = \begin{cases} \max\left(\phi_k - \psi_k, \phi_k^*\right) & \text{if } \phi_k > \phi_k^* \\ \phi_k & \text{otherwise} \end{cases} \tag{28}$$

$$\epsilon_k^* = \epsilon'_k + v_{max}\Delta \tag{29}$$

$$\phi_k^* = \phi'_k + k_1 v_{max}\Delta + k_2\frac{\tan\phi_k}{L} v_{max}\Delta \tag{30}$$

$$\xi_k = -2L \max_{\phi\in[-\phi_k,\phi_k]} \cot\phi \sin\left(-\frac{k_1}{k_2}\epsilon_k^* - \frac{1}{k_2}\phi + \frac{\tan\phi}{2L} v_{max}\Delta\right)\sin\left(\frac{\tan\phi}{2L}\frac{\phi_b}{\delta}\Delta\right) \tag{31}$$

$$\psi_k = \frac{k_2}{L}\tan\phi_k^*\frac{\phi_b}{\delta}\Delta - 2k_1 L\cot\phi_k^*\sin\theta_{k,2}\sin\left(\frac{\tan\phi_k}{2L} v_{max}\Delta\right) \tag{32}$$

$$\epsilon'_k = \max_{\tilde\phi\in[-\phi_k,\phi_k]}\left(-\frac{1}{k_1}\tilde\phi + \frac{k_2}{k_1}\frac{\tan\tilde\phi}{2L} v_{max}\Delta\right) \tag{33}$$

$$\phi'_k = \max\left(\tan^{-1}\sqrt{\frac{2k_1 L^2\delta}{k_2\phi_b\Delta}\sin\theta_{k,2}\sin\left(\frac{\tan\phi_k}{2L} v_{max}\Delta\right)}, \Delta_\phi\right) \tag{34}$$

*where $\varphi$ is the minimum value of $\phi_{k+1}$ such that $\epsilon'_{k+1}$ and $\phi_{k+1}$ satisfy Assumption 5.1(c).*

PROOF. Since by definition $\epsilon_{k+1} \geq \epsilon'_{k+1}$ and $\phi_{k+1} \geq \phi'_{k+1}$, we see that if $|\tau.\mathsf{lstate} \lceil e_1| \leq \epsilon'_{k+1}$ and $|k_1(\tau.\mathsf{lstate} \lceil e_1) + k_2(\tau.\mathsf{lstate} \lceil e_2)| \leq \phi'_{k+1}$, then $\tau.\mathsf{lstate} \in \mathcal{I}_{k+1}$. To show that $\epsilon_{k+1}$ and $\phi_{k+1}$ satisfy Assumption 5.1 and that $a_k, b_k \geq 0$, we use the following observations: (a) $\psi_k \geq 0$ and $\xi_k \geq 0$ and thus, $\epsilon'_{k+1} \leq \epsilon_k$ and $\phi'_{k+1} \leq \phi_k$, (b) given $\phi'_{k+1}$, $\frac{1}{k_1}\phi'_{k+1}$ is the minimum value of $\epsilon_{k+1}$ such that $\epsilon_{k+1}$ and $\phi'_{k+1}$ satisfies Assumption 5.1, (c) given $\epsilon'_{k+1}$, $\varphi$ is the minimum value of $\phi_{k+1}$ such that $\epsilon'_{k+1}$ and $\phi_{k+1}$ satisfies Assumption 5.1, and (d) $\varphi$ decreases as $\epsilon'_{k+1}$ decreases. With these observations and the assumption that $\epsilon_k$ and $\phi_k$ satisfy Assumption 5.1, it can be easily checked that (a) $\epsilon_{k+1} \leq \epsilon_k$ and $\phi_{k+1} \leq \phi_k$, (b) if $\epsilon_k > \epsilon_k^*$ and $\phi_k > \phi_k^*$, then $\epsilon'_{k+1} < \epsilon_k$ and $\phi'_{k+1} < \phi_k$, and (c) if $\epsilon_{k+1} \neq \epsilon'_{k+1}$, then $\phi_{k+1} = \phi'_{k+1}$ and if $\phi_{k+1} \neq \phi'_{k+1}$, then $\epsilon_{k+1} = \epsilon'_{k+1}$. Thus, we can conclude that $\epsilon_{k+1}$ and $\phi_{k+1}$ satisfy Assumption 5.1 and that if $\epsilon_k > \epsilon_k^*$ and $\phi_k > \phi_k^*$, then $\epsilon_{k+1} < \epsilon_k$ and $\phi_{k+1} < \phi_k$.

So what remains to be proved are $|\tau.\mathsf{lstate} \lceil e_1| \leq \epsilon'_{k+1}$ and $|k_1(\tau.\mathsf{lstate} \lceil e_1) + k_2(\tau.\mathsf{lstate} \lceil e_2)| \leq \phi'_{k+1}$. From Theorem 5.6, $\tau.\mathsf{lstate} \in \mathcal{I}_k$. Thus, we can conclude that $\phi'_{k+1} \leq \phi_k$ and $\epsilon'_{k+1} \leq \epsilon_k$. This completes the proof for the second case of (27) and (28).

Next, we prove the first case of (28). Let $\phi_f = -k_1 e_1(0) - k_2 e_2(0)$ and $\phi_l = -k_1 e_1(\Delta) - k_2 e_2(\Delta)$. Suppose $|\phi_f| \geq \Delta_\phi$. From (22), we get that

$$\phi_l \;=\; -k_1 \left(e_1(0) + L\cot\phi_1 \cos(e_2(0)) - L\cot\phi_1 \cos(e_2(\Delta))\right) - k_2 \left(e_2(0) + \frac{\tan\phi_f}{L} v_{avg}\Delta\right)$$

where $v_{avg}$ is the average speed of the vehicle over $\tau$. Substituting $e_1(0) = -\frac{k_2}{k_1}e_2(0) - \frac{1}{k_1}\phi_f$, we get

$$\phi_l \;=\; \phi_f - \left(\frac{k_2}{L}\tan\phi_f v_{avg}\Delta + 2k_1 L\cot\phi_f \sin(\frac{1}{2}(e_2(0)+e_2(\Delta)))\sin\left(\frac{\tan\phi_f}{2L}v_{avg}\Delta\right)\right).$$

Since $\tau.\mathsf{fstate}, \tau.\mathsf{lstate} \in \mathcal{I}_k$, from the definition of $\theta_{k,2}$, we see that $|e_2(0)|, |e_2(\Delta)| \leq \theta_{k,2}$. So $\frac{1}{2}|e_2(0) + e_2(\Delta)| \leq \theta_{k,2}$. In addition, from Theorem 5.6 and the definition of $F_5$ and $F_6$, we know that $\frac{\phi_b}{\delta} \leq v_{avg} \leq v_{max}$. From Lemma 5.8, we get that $\phi_f$ and $\phi_l$ have the same sign. So it is easy to show that

$$|\phi_l| \;\leq\; |\phi_f| - \left(\frac{k_2}{L}\tan|\phi_f|\frac{\phi_b}{\delta}\Delta - 2k_1 L\cot|\phi_f|\sin\theta_{k,2}\sin\left(\frac{\tan\phi_k}{2L}v_{max}\Delta\right)\right).$$

Define the function $\Psi : [0, \phi_k] \to \mathbb{R}$ by

$$\Psi(\phi) \;=\; \frac{k_2}{L}\tan\phi\frac{\phi_b}{\delta}\Delta - 2k_1 L\cot\phi\sin\theta_{k,2}\sin\left(\frac{\tan\phi_k}{2L}v_{max}\Delta\right).$$

That is $\psi_k = \Psi(\phi_k^*)$. It can be easily checked that with Assumption 5.1(e), $\Psi(\phi)$ increases with $\phi$ and vanishes when $\phi = \tan^{-1}\sqrt{\frac{2k_1L^2\delta}{k_2\phi_b\Delta}\sin\theta_{k,2}\sin\left(\frac{\tan\phi_k}{2L}v_{max}\Delta\right)}$ which does not exceed $\phi'_k$ defined in (34). For $\phi > \phi'_k$, $\Psi(\phi) > 0$. From Lemma

5.7, we also know that for any $\phi_f \in [-\phi_k, \phi_k]$,

$$|\phi_l| \;\leq\; |\phi_f| + k_1 v_{max}\Delta + k_2 \frac{\tan\phi_k}{L} v_{max}\Delta.$$

Since $\phi_k^* > \phi_k'$, we arrive at the following conclusion:

$$|\phi_l| \leq \begin{cases} |\phi_f| - \psi_k & \text{if } |\phi_f| > \phi_k^* \\ \phi_k^* & \text{if } \phi_k' \leq |\phi_f| \leq \phi_k^* \\ |\phi_f| + k_1 v_{max}\Delta + k_2 \frac{\tan\phi_k}{L} v_{max}\Delta & \text{if } |\phi_f| < \phi_k' \end{cases}$$

Thus, $|\phi_l| \leq \max(\phi_k - \psi_k, \phi_k^*)$.

Finally, we prove the first case of (27). From (22), we get that

$$e_1(\Delta) \;=\; e_1(0) + 2L\cot\phi_1 \sin\left(e_2(0) + \frac{\tan\phi_f}{2L} v_{avg}\Delta\right) \sin\left(\frac{\tan\phi_f}{2L} v_{avg}\Delta\right).$$

Note that the case where $\phi_f = 0$ is also captured by this equation as $\lim_{\phi_f \to 0} 2L\cot\phi_f \sin\left(\frac{\tan\phi_f}{2L} v_{avg}\Delta\right) = v_{avg}\Delta$. Define the function $\Xi : [0, \epsilon_k] \to \mathbb{R}$ by

$$\Xi(\epsilon) \;=\; -2L \max_{\phi \in [-\phi_k, \phi_k]} \cot\phi \sin\left(-\frac{k_1}{k_2} e - \frac{1}{k_2}\phi + \frac{\tan\phi}{2L} v_{max}\Delta\right) \sin\left(\frac{\tan\phi}{2L}\frac{\phi_b}{\delta}\Delta\right).$$

That is $\xi_k = \Xi(\epsilon_k^*)$. It can be easily checked that with Assumption 5.1(e), $\Xi(\epsilon) > 0$ for any $\epsilon > \epsilon_k'$ and that if $e_1(0) \geq \epsilon_k'$, then $e_2(0) \leq -\frac{k_1}{k_2}\epsilon_k' - \frac{1}{k_2}\phi_f$ and so $2L\cot\phi_f \sin\left(e_2(0) + \frac{\tan\phi_f}{2L} v_{avg}\Delta\right) \sin\left(\frac{\tan\phi_f}{2L} v_{avg}\Delta\right) \leq -\xi_k$. Using symmetry, we can derive similar lower bound for the case where $e_1(0) \leq -\epsilon_k'$. From Lemma 5.7, we also know that

$$|e_1(\Delta)| \;\leq\; |e_1(0)| + v_{max}\Delta$$

So we arrive at the following conclusion:

$$|e_1(\Delta)| \leq \begin{cases} |e_1(0)| - \xi_k & \text{if } |e_1(0)| > \epsilon_k^* \\ \epsilon_k^* & \text{if } \epsilon_k' \leq |e_1(0)| \leq \epsilon_k^* \\ |e_1(0)| + v_{max}\Delta & \text{if } |e_1(0)| < \epsilon_k' \end{cases}$$

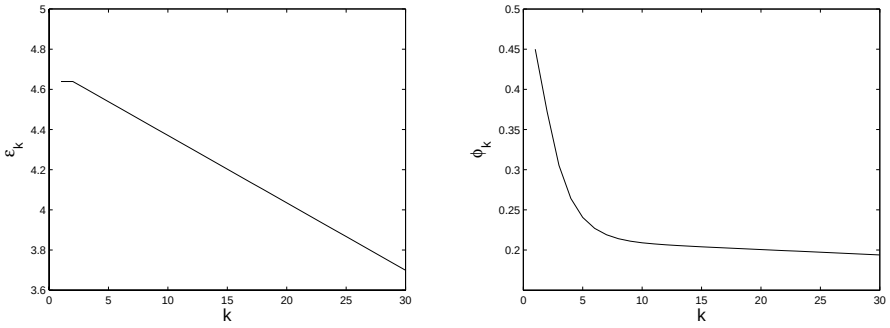
Thus, $|e_1(\Delta)| \leq \max(\epsilon_k - \xi_k, \epsilon_k^*)$. □

Define $k^*$ to be the minimum value of $k$ such that $\epsilon_k \leq \epsilon_k^*$ or $\phi_k \leq \phi_k^*$. (If for any $k$, $\epsilon_k > \epsilon_k^*$ and $\phi_k > \phi_k^*$, just pick an arbitrary natural number $k^*$.) Then, for any $k < k^*$, $a_k$ and $b_k$ are strictly positive, that is, $I_{k+1} \subsetneq I_k$. The plot showing the progress in the deviation and disoriantation is shown in Figure 8.

The following lemma provides the value of the bound $d^*$ on the maximum change in the value of $d$ over $\tau$

**Lemma 5.9.** *Suppose $\tau.\mathsf{fstate} \in \mathcal{I}_k$ for some $k \in \mathbb{N}$. For any $t \in dom(\tau)$, $|(\tau \lceil d)(t) - \tau.\mathsf{fstate} \lceil d| \leq d^*$ where $d^* = v_{max}\Delta$.*

Proof. From Theorem 5.6, the definition of $F_5$ and $F_6$ and the definition of $f_7$ which describes the evolution of $d$, we get that $\max_{s,s_0 \in I} \|f_7(s, g(l, s_0))\| \leq v_{max}$.




(a) (b)

Fig. 8. The progress in deviation and disorientation. (a) The relationship between $\epsilon_k$ and $k$. (b) The relationship between $\phi_k$ and $k$

Since $dom(\tau) = [0, \Delta]$, we get $|(\tau \downarrow d)(t) - \tau.\mathsf{fstate} \lceil d| \leq \max_{s,s_0 \in I} \|f_7(s, g(l, s_0))\|\Delta \leq v_{max}\Delta$. □

Using Lemma 5.8 and Lemma 5.9, we establish the relationship between the progress of $\mathcal{I}_k$'s and the decrease in the value of $d$.

**Lemma 5.10.** *For each $k \in \mathbb{N}$, starting from any reachable state $\mathbf{x} \in \mathcal{I}_k$ such that $\mathbf{x}.d > v_{max}\Delta$, $\mathbf{x}.path = \mathbf{x}.new\_path$ and $\mathbf{x}.next = \mathbf{x}.now$, any* plan*-free execution fragment $\beta$ with $\beta.\mathsf{ltime} = \Delta$ satisfies $\beta.\mathsf{lstate} \in \mathcal{I}_{k+1}$ and $\beta.\mathsf{lstate} \lceil d \geq \mathbf{x}.d - v_{max}\Delta$.*

Proof. Since $\mathbf{x}.next = \mathbf{x}.now$ and $\beta.\mathsf{ltime} = \Delta$, we see that $\beta$ can be written as $\beta = \beta'$ or $\beta = \beta'\mathsf{main}\tau_j\mathsf{brake}(b_j)\tau_{j+1}\mathsf{brake}(b_{j+1})\ldots\tau_n$ where $\beta'$ is an execution fragment with exactly one main action $a_i$ which occurs at time 0 and is immediately followed by a main action in the execution, $\beta'.\mathsf{ltime} = \Delta$ and $\tau_j, \ldots \tau_n$ are point trajectories. Let $\tau$ be the pasted trajectory of all the trajectories after $a_i$ in $\beta'$. Then, $\tau$ is a pasted trajectory of all the trajectories between two main actions and so Lemma 5.8 and Lemma 5.9 apply. Since the main action $a_i$ occurs at time 0 in $\beta$ and brake action does not affect the value of $s$, we see that $\tau_{i-1}.\mathsf{lstate} \lceil s = \mathbf{x}.s$. So $\tau_{i-1}.\mathsf{lstate} \lceil d > v_{max}\Delta > 0$ and hence $a_i$ does not change the value of $s$. That is, $\tau.\mathsf{fstate} = \mathbf{x} \in \mathcal{I}_k$. From Lemma 5.8, we get that $\beta'.\mathsf{lstate} \in \mathcal{I}_{k+1}$. In addition, from Lemma 5.9, we see that $\beta'.\mathsf{lstate} \lceil d \geq \mathbf{x}.d - v_{max}\Delta$. Since $\mathbf{x}.d > v_{max}\Delta$, we get $\beta'.\mathsf{lstate} \lceil d > 0$. Therefore, the main action following $\beta'$ does not change the value of $s$. In addition, since brake action only affects the *brake* variable, we see that $\beta.\mathsf{lstate} \lceil s = \beta'.\mathsf{lstate} \lceil s$. Hence, we can conclude that $\beta.\mathsf{lstate} \in \mathcal{I}_{k+1}$ and $\beta.\mathsf{lstate} \lceil d \geq \mathbf{x}.d - v_{max}\Delta$. □

Finally, we conclude the section by establishing the segment progress property (B) defined at the beginning of Section 5.

**Theorem 5.11.** *For each $k \in \mathbb{N}$, starting from any reachable state $\mathbf{x} \in \mathcal{I}_k$, any reachable state $\mathbf{x}'$ is in $\mathcal{I}_{k+n}$ where $n = \max(\lfloor \frac{\mathbf{x}.d - \mathbf{x}'.d}{v_{max}\Delta} \rfloor - 1, 0)$, provided that path and current segment do not change.*

PROOF. Consider an arbitrary closed execution fragment $\beta$ starting at $\mathbf{x}$ and ending at $\mathbf{x}'$. Since by assumption, $\beta$ is a plan-free execution fragment such that $\beta.\mathsf{lstate} \lceil path = \beta.\mathsf{fstate} \lceil new\_path$ and $\beta.\mathsf{lstate} \lceil seg = \beta.\mathsf{fstate} \lceil seg$, from Theorem 5.6, we know that $\beta.\mathsf{lstate} \in \mathcal{I}_k$. This completes the proof for the case where $\lfloor \frac{\mathbf{x}.d - \mathbf{x}'.d}{v_{max}\Delta} \rfloor - 1 \leq 0$.

Next, consider the case where $\lfloor \frac{\mathbf{x}.d - \mathbf{x}'.d}{v_{max}\Delta} \rfloor - 1 > 0$. From the structure of a PCHA, we see that $next = now$ every $\Delta$ time. So, the first state in $\beta$ such that $next = now$ occurs no later than time $\Delta$. Using Lemma 5.9, we see that at this state, $d \geq \mathbf{x}.d - v_{max}\Delta$. Applying Lemma 5.10 and using an invariance of $\mathcal{I}_k$ for any $k$ proved in Theorem 5.6, we get that $\beta_1.\mathsf{lstate} \in \mathcal{I}_{k+n}$ where $n = \lfloor \frac{\mathbf{x}.d - v_{max}\Delta - \mathbf{x}'.d}{v_{max}\Delta} \rfloor$. □

A sequence of shrinking $\mathcal{I}_k$'s visited by $\mathcal{A}$ in making progress towards a waypoint is shown in Figure 9.

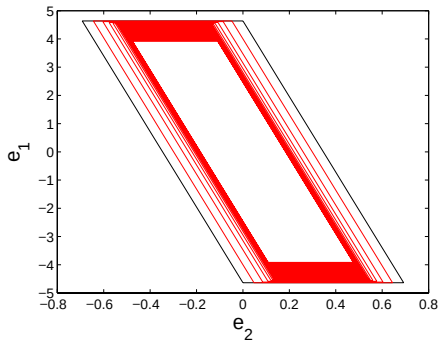



Fig. 9. A sequence of shrinking $\mathcal{I}_k$'s visited by $\mathcal{A}$ in making progress towards a waypoint. $\mathcal{I}_k$ is drawn in black, whereas $\mathcal{I}_{k+i}$ is drawn in red for $i > 0$.

### 5.4 Safety and Waypoint Progress: Identifying Safe Planner Paths

In this section, we derive a sufficient condition on planner paths that can be safely followed with respect to a candidate invariant set $\mathcal{I}_0$ whose parameters $\epsilon_0 \in [0, e_{max}]$ and $\phi_0 \in [0, \phi_{max}]$ satisfy Assumption 5.1 and are chosen such that $\mathcal{I}_0$ contains the initial state $Q_{0\mathcal{A}}$. Then, we prove an invariance of $\mathcal{I}_0$ and conclude that the safety and waypoint progress properties (A) and (C) defined at the beginning of Section 5 are satisfied.

The proof is structured as follows. First, we consider an execution fragment where path does not change and starting with waypoint-distance not shorter than some threshold $D^*$. Lemma 5.15 uses the segment progress property established in Section 5.3 to prove that this execution fragment preserves an invariance of $\mathcal{I}_0$. Then, in Lemma 5.16 and Lemma 5.17, we show that right after a path is changed, the waypoint-distance is not shorter than $D^*$ and the state of $\mathcal{A}$ remains in $\mathcal{I}_0$. Using these results, Lemma 5.18 concludes that an execution fragment which updates the path exactly once by the first main action preserves an invariance of $\mathcal{I}_0$. Finally, we use Lemma 5.15 and Lemma 5.18 to conclude the section that $\mathcal{I}_0$ is in fact an invariant of $\mathcal{A}$ and with this result, we conclude that the system satisfies

the safety and waypoint progress properties (A) and (C) defined at the beginning of Section 5.

The following assumption provides sufficient conditions for *planner* paths that can be safely followed. The key idea in the condition is: *longer path segments can be succeeded by sharper turns*. Following a long segment, the vehicle reduces its deviation and disorientation by the time it reaches the end, and thus, it is possible for the vehicle to turn more sharply at the end without breaking an invariance of $\mathcal{I}_0$.

***Assumption* 5.12.** (Planner paths) Let $p_0, p_1, \ldots$ be a planner path; for $i \in \{0, 1, \ldots\}$, let $\lambda_i$ be the length of the segment $\overline{p_i p_{i+1}}$ and $\sigma_i$ be the difference in orientation of $\overline{p_i p_{i+1}}$ and that of $\overline{p_{i+1} p_{i+2}}$. Then, for each $i \in \{0, 1, \ldots\}$,

(a) $\lambda_i \geq 2v_{max}\Delta + \epsilon_0$.

(b) Let $n = \lfloor \frac{\lambda_i - \epsilon_0 - 2v_{max}\Delta}{v_{max}\Delta} \rfloor$. Then, $\lambda_i$ and $\sigma_i$ satisfy the following conditions:

$$\epsilon_n \leq \frac{1}{|\cos\sigma_i|}(\epsilon_0 - v_{max}\Delta|\sin\sigma_i|) \tag{35}$$

$$\phi_n \leq \phi_0 - k_1 v_{max}\Delta \sin|\sigma_i| - k_1\epsilon_n(1 - \cos\sigma_i) - k_2|\sigma_i| \tag{36}$$

where, given $\epsilon_0$ and $\phi_0$, $\epsilon_n$ and $\phi_n$ are defined recursively for any $n > 0$ by $\epsilon_n = \epsilon_{n-1} - a_{n-1}$ and $\phi_n = \phi_{n-1} - b_{n-1}$ where $a_{n-1}, b_{n-1}$ are defined in Lemma 5.8.

The relationship between $\lambda$ and the maximum value of $\sigma$ which satisfies this assumption is shown in Figure 10.

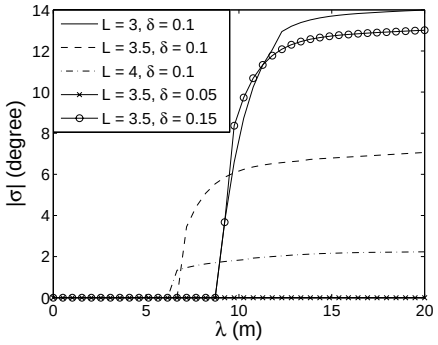



Fig. 10. Segment length vs. maximum difference between consecutive segment orientations, for different values of $L$ and $\delta$.

***Remark* 5.13.** The choice of $\epsilon_0$'s and $\phi_0$'s affects both the requirements on a safe path (Assumption 5.12) and the definition of a *good* brake controller (Definition 5.2). Larger $\epsilon_0$'s and $\phi_0$'s allow sharper turns in planned paths but forces brakes to occur only at higher speeds. That is, relaxing the constraint on a path results in the tighter constraint on a brake action. This tradeoff is related to the design flaw of Alice as discussed in the introduction of the paper. Without having

quantified the tradeoff, we inadvertently allowed a path to have sharp turns and also brakes at low speeds—thus violating safety.

To establish that $\mathcal{I}_0$ is an invariant of $\mathcal{A}$, we further assume that (a) new planner paths begin at the current position, (b) Vehicle is not too disoriented with respect to new paths, and (c) Vehicle speed is not too high as stated in Assumption 5.14.

***Assumption* 5.14.** (plan action and new path)

(a) Any new path $p = p_1 p_2 \ldots$ satisfies $p_1 = [x_p, y_p]$ where $x_p$ and $y_p$ are the values of the variable $x$ and $y$, respectively, when the path is received (i.e. when the plan action occurs). That is, for any new input path, the path must begin at the current position of the vehicle.

(b) Let $v_p$ and $\theta_p$ be the speed and the orientation of the vehicle, respectively, when a plan action occurs. Then,

$$v_p < \frac{\epsilon_0}{\Delta\sqrt{1+\sin^2\theta_{0,2}}} - a_{max}\Delta$$

where given $\epsilon_0$ and $\phi_0$, $\theta_{0,2}$ is defined as in (16). In addition, let $p = p_1 p_2 \ldots$ be the received path and let $\vec{p}$ be the vector which represents a straight line defined by $p_1$ and $p_2$. Then,

$$|\angle\vec{p} - \theta_p| \leq \frac{\phi_0}{k_2} - (v_p + a_{max}\Delta)\Delta\left(\frac{k_1}{k_2}\sqrt{1+\sin^2\theta_{0,2}} + \frac{\tan\phi_0}{L}\right).$$

First, we consider an execution fragment where path does not change and starting with a large enough waypoint-distance. The following lemma uses the progress property established in Section 5.3 to shows that before switching to the next segment, $\mathbf{x} \in \mathcal{I}_n$ where $n \geq 0$ depends on the segment length. Since we restrict the sharpness of the turn with respect to segment length (Assumption 5.12), we can then conclude that this execution fragment preserves an invariance of $\mathcal{I}_0$.

**Lemma 5.15.** *Consider a* plan*-free execution fragment $\beta$ starting at a state $\mathbf{x} \in \mathcal{I}_0$. Suppose $\mathbf{x}.path = \mathbf{x}.new\_path$ and $\mathbf{x}.d \geq D^*$ where $D^* = \lambda_1 - \epsilon_0 - v_{max}\Delta$ and $\lambda_1$ is the length of the segment $\mathbf{x}.seg$. Then $\beta.\mathsf{lstate} \in \mathcal{I}_0$.*

Proof. First, observe that $\beta$ can be written as $\beta = \beta_1 a_1 \beta_2 a_2 \ldots \beta_m$ where for any $i$, $a_i$ is a main action and $\beta_i$ is a plan-free execution fragment such that $\beta_i.\mathsf{lstate} \lceil path = \beta_i.\mathsf{fstate} \lceil new\_path$ and $\beta_i.\mathsf{lstate} \lceil seg = \beta_i.\mathsf{fstate} \lceil seg$. From Theorem 5.6, we get that for any $i$, if $\beta_i.\mathsf{fstate} \in \mathcal{I}_0$, then $\beta.\mathsf{lstate} \in \mathcal{I}_0$. So, suppose $\beta_1.\mathsf{fstate} \in \mathcal{I}_0$, $\beta_1.\mathsf{fstate} \lceil path = \beta_1.\mathsf{fstate} \lceil new\_path$ and $\beta_1.\mathsf{fstate} \lceil d \geq \lambda_1 - \epsilon_0 - v_{max}\Delta$. We only need to show that for any $i > 1$, $\beta_i.\mathsf{fstate} \in \mathcal{I}_0$.

Consider the base case $i = 2$. If $\beta_2.\mathsf{fstate} \lceil seg = \beta_1.\mathsf{lstate} \lceil seg$, then $a_1$ does not change the continuous state $s$, and so $\beta_2.\mathsf{fstate} \in \mathcal{I}_0$. Otherwise, $\beta_2.\mathsf{fstate} \lceil seg = \beta_1.\mathsf{fstate} \lceil seg + 1$. But from the update rule of the variable $seg$ and Lemma 5.9, it can be easily shown that $-v_{max}\Delta < \beta_1.\mathsf{lstate} \lceil d \leq 0$. Applying Theorem 5.11, we get that $\beta_1.\mathsf{lstate} \in \mathcal{I}_n$ where $n = \lfloor \frac{\lambda_1 - \epsilon_0 - 2v_{max}\Delta}{v_{max}\Delta} \rfloor$ because by Assumption 5.12(a), $\lambda_1 - \epsilon_0 - 2v_{max}\Delta > 0$.

Let $\mathbf{x}_1 = \beta_1.\mathsf{lstate}$ and $\mathbf{x}_2 = \beta_2.\mathsf{fstate}$ and let $\sigma_1$ be the difference between the orientation of $\beta_1.\mathsf{fstate} \lceil seg$ and $\beta_1.\mathsf{fstate} \lceil seg+1$. From the update rule for $e_1$ and

the definition of $\vec{p}$, $\vec{q}$ and $\vec{r}$ in Figure 5, it can be shown that $\mathbf{x}_2.e_1 = \mathbf{x}_1.d \sin\sigma_1 + \mathbf{x}_1.e_1 \cos\sigma_1$. But since $\beta_1.\mathsf{lstate} \in \mathcal{I}_n$, from the definition of $\mathcal{I}_n$, $|\mathbf{x}_1.e_1| \leq \epsilon_n$. Therefore, using the bounds on $\mathbf{x}_1.d$ provided earlier in the proof, we get $|\mathbf{x}_2.e_1| \leq v_{max}\Delta|\sin\sigma_1| + \epsilon_n|\cos\sigma_1|$. Hence, from Assumption 5.12(b), $|\mathbf{x}_2.e_1| \leq \epsilon_0$, that is, $F_1(\mathbf{x}_2.s), F_2(\mathbf{x}_2.s) \geq 0$.

Next, we prove that $F_3(\mathbf{x}_2.s), F_4(\mathbf{x}_2.s) \geq 0$. From the definition of $\mathcal{I}_n$, we know that $-\frac{k_1}{k_2}\mathbf{x}_1.e_1 - \frac{1}{k_2}\phi_n \leq \mathbf{x}_1.e_2 \leq -\frac{k_1}{k_2}\mathbf{x}_1.e_1 + \frac{1}{k_2}\phi_n$. From the update rule for $e_2$, it can be easily shown that $\mathbf{x}_2.e_2 = \mathbf{x}_1.e_2 - \sigma_1$. Thus, we get that $-\frac{k_1}{k_2}\mathbf{x}_1.e_1 - \frac{1}{k_2}\phi_n - \sigma_1 \leq \mathbf{x}_2.e_2 \leq -\frac{k_1}{k_2}\mathbf{x}_1.e_1 + \frac{1}{k_2}\phi_n - \sigma_1$. Using the bounds on $\mathbf{x}_2.e_1$, $\mathbf{x}_2.e_2$ and $\mathbf{x}_1.d$, we can derive that $k_1\mathbf{x}_2.e_1 + k_2\mathbf{x}_2.e_2 \leq k_1 v_{max}\Delta \sin|\sigma_1| + k_1\epsilon_n(1-\cos\sigma_1) + \phi_n + k_2|\sigma_1|$ and $k_1\mathbf{x}_2.e_1 + k_2\mathbf{x}_2.e_2 \geq -k_1 v_{max}\Delta \sin|\sigma_1| - k_1\epsilon_n(1-\cos\sigma_1) - \phi_n - k_2|\sigma_1|$. That is,

$$|k_1\mathbf{x}_2.e_1 + k_2\mathbf{x}_2.e_2| \;\leq\; k_1 v_{max}\Delta \sin|\sigma_1| + k_1\epsilon_n(1-\cos\sigma_1) + \phi_n + k_2|\sigma_1|$$

Therefore, Assumption 5.12(b) guarantees that $|k_1\mathbf{x}_2.e_1 + k_2\mathbf{x}_2.e_2| \leq \phi_0$. That is, $F_3(\mathbf{x}_2.s), F_4(\mathbf{x}_2.s) \geq 0$. In addition, since a $\mathsf{main}$ action does not affect $v$, $F_5(\mathbf{x}_2.s) = F_5(\mathbf{x}_1.s)$ and $F_6(\mathbf{x}_2.s) = F_6(\mathbf{x}_1.s)$, so $F_5(\mathbf{x}_2.s), F_6(\mathbf{x}_1.s) \geq 0$.

Therefore, by definition of $\mathcal{I}_0$, we get $\beta_2.\mathsf{fstate} \in \mathcal{I}_0$. In addition, from the bounds on $\mathbf{x}_1.d$ and $\mathbf{x}_1.e_1$, it can be easily shown that $\beta_2.\mathsf{fstate} \lceil d \geq \lambda_2 - \epsilon_0 - v_{max}\Delta$ where $\lambda_2$ is the length of the segment $\beta_2.\mathsf{fstate} \lceil seg$.

Next, consider an arbitrary $i \geq 2$ and assume that $\beta_{i-1}.\mathsf{fstate} \in \mathcal{I}_0$ and if $i = 2$ or $i > 2$ and $\beta_{i-1}.\mathsf{fstate} \lceil seg \neq \beta_{i-2}.\mathsf{lstate} \lceil seg$, then $\beta_{i-1}.\mathsf{fstate} \lceil d \geq \lambda_{i-1} - \epsilon_0 - v_{max}\Delta$ where $\lambda_{i-1}$ is the length of the segment $\beta_{i-1}.\mathsf{fstate} \lceil seg$. Simply following the previous proof for $i = 2$, we get $\beta_i.\mathsf{fstate} \in \mathcal{I}_0$ and if $\beta_i.\mathsf{fstate} \lceil seg \neq \beta_{i-1}.\mathsf{lstate} \lceil seg$, then $\beta_i.\mathsf{fstate} \lceil d \geq \lambda_i - \epsilon_0 - v_{max}\Delta$ where $\lambda_i$ is the length of the segment $\beta_i.\mathsf{fstate} \lceil seg$.

By mathematical induction, we conclude the proof that for any $i > 1$, $\beta_i.\mathsf{fstate} \in \mathcal{I}_0$. □

The next two lemmas show that Assumption 5.14 is sufficient to guarantee that if a path is changed, then all the assumptions in the Lemma 5.15 are satisfied.

**Lemma 5.16.** *For each state $\mathbf{x}, \mathbf{x}' \in Q$ such that $\mathbf{x}.path \neq \mathbf{x}.new\_path$, if $\mathbf{x} \in \mathcal{I}_0$ and $\mathbf{x} \overset{\mathsf{main}}{\rightarrow} \mathbf{x}'$, then $\mathbf{x}'.d \geq \lambda - v_{max}\Delta > 0$ where $\lambda$ is the length of the first segment of $\mathbf{x}.new\_path$.*

Proof. Consider an arbitrary execution $\alpha = \tau_0 a_1 \tau_1 a_2 \ldots$. Pick an arbitrary natural number $i$ such that $a_i$ is a $\mathsf{main}$ action and let $\mathbf{x} = \tau_{i-1}.\mathsf{lstate}$ and $\mathbf{x}' = \tau_i.\mathsf{fstate}$. We want to show that if $\mathbf{x} \lceil path \neq \mathbf{x} \lceil new\_path$, then $\mathbf{x}'.d \geq \lambda - v_{max}\Delta > 0$. Notice that $\mathbf{x}.path \neq \mathbf{x}.new\_path$ if and only if there exists a natural number $j < i$ such that $a_j$ is a $\mathsf{plan}$ action and for any natural number $k \in \{j+1, \ldots, i-1\}$, $a_k$ is not a $\mathsf{main}$ action. Using Assumptions 5.14(a), we get $\langle \tau_j.\mathsf{fstate} \lceil x, \tau_j.\mathsf{fstate} \lceil y \rangle = p_{i,1}$ where $p_{i,1}$ is the first waypoint in $\mathbf{x}.new\_path$. Since $\mathsf{main}$ action occurs every $\Delta$ time, the time between $a_i$ and $a_j$ is at most $\Delta$. Therefore, from Theorem 5.6, the definition of $F_5$ and $F_6$ and the definition of $f_1$ and $f_2$ which describe the evolution of $x$ and $y$, we see that $\|\langle \mathbf{x}.x, \mathbf{x}.y \rangle - p_{i,1}\| \leq v_{max}\Delta$. Furthermore, from Assumption 5.12(a), we know that $\lambda = \|p_{i,2} - p_{i,1}\| > v_{max}\Delta + \epsilon_0$

where $p_{i,2}$ is the second waypoint in $p_i$. Thus, $\mathbf{x}.d \geq \|p_{i,2} - p_{i,1}\| - \|\langle \mathbf{x}.x, \mathbf{x}.y\rangle - p_{i,1}\| \geq \lambda - v_{max}\Delta > 0$. □

**Lemma 5.17.** *For each state $\mathbf{x}, \mathbf{x}' \in Q$ such that $\mathbf{x}.path \neq \mathbf{x}.new\_path$, if $\mathbf{x} \in \mathcal{I}_0$ and $\mathbf{x} \overset{\mathsf{main}}{\rightarrow} \mathbf{x}'$, then $\mathbf{x}' \in \mathcal{I}_0$.*

PROOF. Consider an arbitrary execution $\alpha = \tau_0 a_1 \tau_1 a_2 \ldots$. Pick an arbitrary natural number $i$ such that $a_i$ is a main action and let $\mathbf{x} = \tau_{i-1}.\mathsf{lstate}$ and $\mathbf{x}' = \tau_i.\mathsf{fstate}$. We want to show that if $\mathbf{x} \in \mathcal{I}_0$ and $\mathbf{x}.path \neq \mathbf{x}.new\_path$, then $\mathbf{x}' \in \mathcal{I}_0$. So suppose $\mathbf{x} \in \mathcal{I}_0$. Notice that $\mathbf{x}.path \neq \mathbf{x}.new\_path$ if and only if there exists a natural number $j < i$ such that $a_j$ is a plan action and for any natural number $k \in \{j+1, \ldots, i-1\}$, $a_k$ is not a main action. Let $p_{j1}$ and $p_{j2}$ be the first two waypoints of the new path. Consider a closed execution fragment $\beta = \tau_j a_{j+1} \ldots \tau_{i-1}$. From Assumption 5.14(a), we get that $p_{j1} = \tau_j.\mathsf{fstate} \lceil \langle x, y\rangle$. Since main action occurs every $\Delta$ time, we see that $\beta.\mathsf{ltime} \leq \Delta$. From the differential equations describing the evolution of $x$ and $y$, we get that

$$\begin{aligned} |(\tau_j.\mathsf{fstate} \lceil x) - (\mathbf{x}.x)| &\leq ((\tau_j.\mathsf{fstate} \lceil v) + a_{max}\Delta)\Delta \\ |(\tau_j.\mathsf{fstate} \lceil y) - (\mathbf{x}.y)| &\leq \sin\theta_{0,2}((\tau_j.\mathsf{fstate} \lceil v) + a_{max}\Delta)\Delta \end{aligned}$$

So from the definition of $\vec{r}$ in Figure 5, we get that

$$\|\vec{r}\| \leq (\tau_j.\mathsf{fstate} \lceil v) + a_{max}\Delta)\Delta\sqrt{1 + \sin^2\theta_{0,2}}$$

Using Assumption 5.14(b), we can conclude that $\|\vec{r}\| \leq \epsilon_0$. So from the update rule for $e_1$, $|\mathbf{x}'.e_1| \leq \|\vec{r}\|$ and so

$$|\mathbf{x}'.e_1| \leq (\tau_j.\mathsf{fstate} \lceil v) + a_{max}\Delta)\Delta\sqrt{1 + \sin^2\theta_{0,2}} \leq \epsilon_0, \tag{37}$$

that is $F_1(\mathbf{x}'.s), F_2(\mathbf{x}'.s) \geq 0$.

Similarly, from the differential equation describing the evolution of $\theta$, we get that

$$|(\tau_j.\mathsf{fstate} \lceil \theta) - (\mathbf{x}.\theta)| \leq \frac{1}{L}\tan\phi_0((\tau_j.\mathsf{fstate} \lceil v) + a_{max}\Delta)\Delta$$

Using condition (1) of Assumption 5.14(b), we can conclude that

$$\begin{aligned} |\angle\vec{p} - (\mathbf{x}.\theta)| &= |(\angle\vec{p} - (\tau_j.\mathsf{fstate} \lceil \theta)) + ((\tau_j.\mathsf{fstate} \lceil \theta) - (\mathbf{x}.\theta))| \\ &\leq |(\angle\vec{p}_i - (\tau_j.\mathsf{fstate} \lceil \theta))| + |((\tau_j.\mathsf{fstate} \lceil \theta) - (\mathbf{x}.\theta))| \\ &\leq \frac{\phi_0}{k_2} - \frac{k_1}{k_2}((\tau_j.\mathsf{fstate} \lceil v) + a_{max}\Delta)\Delta\sqrt{1 + \sin^2\theta_{0,2}} \end{aligned}$$

So we get

$$|k_2\mathbf{x}'.e_2| \leq \phi_0 - k_1((\tau_j.\mathsf{fstate} \lceil v) + a_{max}\Delta)\Delta\sqrt{1 + \sin^2\theta_{0,2}}$$

Combining this with (37), we get that

$$|k_1(\mathbf{x}'.e_1) + k_2(\mathbf{x}'.e_2)| \leq |k_1(\mathbf{x}'.e_1)| + |k_2(\mathbf{x}'.e_2)| \leq \phi_0,$$

that is, $F_3(\mathbf{x}'.s), F_4(\mathbf{x}'.s) \geq 0$.

In addition, since main action does not affect $v$, we see that $F_5(\mathbf{x}'.s) = F_5(\mathbf{x}.s)$ and $F_6(\mathbf{x}'.s) = F_6(\mathbf{x}.s)$, so $F_5(\mathbf{x}'.s), F_6(\mathbf{x}'.s) \geq 0$. Therefore, by definition of $\mathcal{I}_0$, we get that $\mathbf{x}' \in \mathcal{I}_0$. □

Using the previous three lemmas, the following lemma concludes that an execution fragment which updates the path exactly once by the first main action preserves an invariance of $\mathcal{I}_0$.

**Lemma 5.18.** *Consider a* plan*-free execution fragment $\beta$ starting at a state $\mathbf{x} \in \mathcal{I}_0$. If $\mathbf{x}.path \neq \mathbf{x}.new\_path$, then $\beta.\mathsf{lstate} \in \mathcal{I}_0$.*

PROOF. $\beta$ can be written as $\beta = \beta_1 \mathsf{main} \beta_2$ where $\beta_1 = \tau_0 \mathsf{brake} \tau_1 \mathsf{brake} \ldots \tau_n$ and $\beta_2$ is a plan-free execution fragment with $\beta_2.\mathsf{fstate} \lceil path = \beta_2.\mathsf{fstate} \lceil new\_path$. Clearly, $\beta_1.\mathsf{lstate} \lceil path \neq \beta_1.\mathsf{lstate} \lceil new\_path$. In addition, $\beta_1.\mathsf{fstate} \in \mathcal{I}_0$ and thus, from Theorem 5.6, $\beta_1.\mathsf{lstate} \in \mathcal{I}_0$. Applying Lemma 5.16 and Lemma 5.17, we see that $\beta_2.\mathsf{fstate} \lceil d \geq \lambda_1 - v_{max}\Delta \geq \lambda_1 - \epsilon_0 - v_{max}\Delta$ and $\beta_2.\mathsf{fstate} \in \mathcal{I}_0$ where $\lambda_1$ is the length of the first segment of $\mathbf{x}.new\_path$. Therefore, from Lemma 5.15, $\beta.\mathsf{lstate} \in \mathcal{I}_0$. □

Finally, we conclude that $\mathcal{I}_0$ is an invariant of $\mathcal{A}$.

**Theorem 5.19.** *Suppose the initial state $\mathbf{x}_0 \in \mathcal{I}_0$ and $\mathbf{x}_0.d \geq \lambda_1 - \epsilon_0 - v_{max}\Delta$ where $\lambda_1$ is the length of the first segment of the initial path. Then, $\mathcal{I}_0$ is an invariant of $\mathcal{A}$.*

PROOF. Any execution $\alpha$ can be written as $\alpha = \beta_1 \mathsf{plan} \beta_2 \mathsf{plan} \ldots$ where $\beta_1$ is a plan-free execution fragment with $\beta_1.\mathsf{fstate} \lceil path = \beta_1.\mathsf{fstate} \lceil new\_path$ and for any $i \geq 2$, $\beta_i$ is a plan-free execution fragment with $\beta_i.\mathsf{fstate} \lceil path \neq \beta_i.\mathsf{fstate} \lceil new\_path$. Since plan action does not affect the variable $s$, if $\beta_1.\mathsf{lstate} \in \mathcal{I}_0$, then $\beta_2.\mathsf{fstate} \in \mathcal{I}_0$ and using Lemma 5.18, we get that for any $i \geq 2$, $\beta_i.\mathsf{lstate} \in \mathcal{I}_0$. Thus, we only need to show that $\beta_1.\mathsf{lstate} \in \mathcal{I}_0$. But this is true from Lemma 5.15 since $\beta_1.\mathsf{fstate} \lceil d = \mathbf{x}_0.d \geq \lambda_1 - \epsilon_0 - v_{max}\Delta$ and $\beta_1.\mathsf{fstate} \in \mathcal{I}_0$. □

Since for any state $\mathbf{x} \in \mathcal{I}_0$, $|\mathbf{x}.e_1| \leq \epsilon_0 \leq e_{max}$, invariance of $\mathcal{I}_0$ guarantees the safety property (A). For property (C), we note that for any state $\mathbf{x} \in \mathcal{I}_0$, there exists $v_{min} > 0$ such that $\mathbf{x}.v \geq v_{min} > 0$ and $|\mathbf{x}.e_2| \leq \theta_{0,2} < \frac{\pi}{2}$, that is, $\dot{d} = f_7(\mathbf{x}.s, u) \leq -v_{min} \cos\theta_{0,2} < 0$ for any $u \in \mathcal{U}$. Thus, it follows that the waypoint distance decreases and the vehicle makes progress towards its waypoint.

The simulation results are shown in Figure 11 which illustrate that the vehicle is capable of making a sharp left turn, provided that the path satisfies Assumption 5.12. In addition, we are able to replicate the stuttering behavior described in the Introduction when Assumption 5.12 is violated.

## 6. CONCLUSIONS

Motivated by a design bug that caused an undesirable behavior of Alice, an autonomous vehicle built at Caltech for the 2007 DARPA Urban Challenge, this paper introduced Periodically Controlled Hybrid Automata (PCHA), a subclass of Hybrid I/O Automata that is suitable for modeling sampled control systems and embedded systems with periodic sensing and actuation. New sufficient conditions for verifying invariant properties of PCHAs were presented. These conditions can

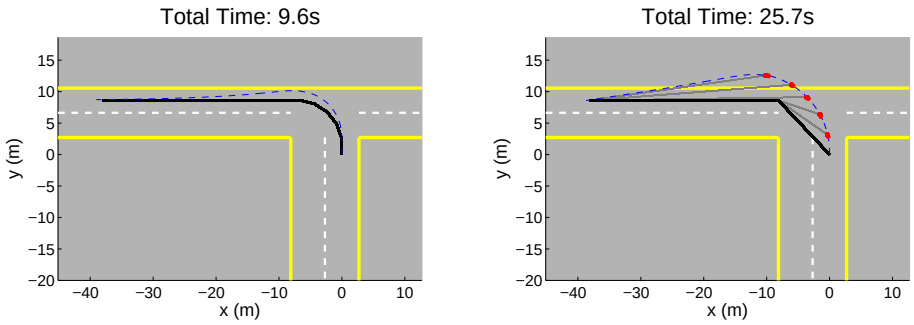



Fig. 11. The positions of the vehicle as it follows a path to execute a sharp left turn. The solid line and the dashed line represent, respectively, the path and the positions of the vehicle. The initial path is drawn in thick solid (black) line. The positions of the vehicle are plotted in thin dashed (blue) line except when *brake* is triggered in which case they are plotted in thick dashed (red) line. *Left.* The path satisfies Assumption 5.12. *Right.* The path does not satisfy Assumption 5.12 and the replan occurs due to excessive deviation. The replanned paths are drawn in thin solid (grey) line.

be automatically checked using, for example, the constraint-based approach, quantifier elimination, or sum of squares decomposition. The intuition behind these conditions is that for an execution fragment to leave an invariant set $\mathcal{I}$, it needs to cross the boundary $\partial\mathcal{I}$ of $\mathcal{I}$. Hence, to verify invariance of $\mathcal{I}$, it is suffice to identify a subset $C$ of $\mathcal{I}$ such that: (1) there is enough separation between $C$ and $\partial\mathcal{I}$ to ensure that if a control law is evaluated when the state is inside $C$, then it is evaluated again before an execution fragment reaches $\partial\mathcal{I}$, and (2) if the control law is evaluated when the state is outside $C$, then the vector field on $\partial\mathcal{I}$ points inwards with respect to $\partial\mathcal{I}$. These conditions can be generalized to the case where a collection of subsets $C$'s corresponding to different parts of $\partial\mathcal{I}$ is needed to prove invariance of $\mathcal{I}$. An example presented in the paper describes how an invariant set can be automatically determined using the constraint-based approach.

We then applied the proposed technique to verify a sequence of invariant properties of the planner-controller subsystem of Alice. Geometric properties of planner generated paths are derived which guarantee that such paths can be safely followed by the controller. The analysis revealed that the software design was not inherently flawed; the undesirable behavior was caused by an unfortunate choice of certain parameters. The simulation results verified that with the proper choice of parameters, the observed failure does not occur.

An interesting direction for future research is towards automatic invariant proofs of PCHAs combining the proofs for invariance of control steps and for invariance of control-free fragments based on the results of Lemma 3.1. Invariance of control steps can be partially automated using a theorem prover (e.g. PVS [Owre et al. 1996]) while invariance of control-free fragments can be automated using software tools for solving sum of squares problems (e.g. SOSTOOLS [Prajna et al. 2002]) or software tools for quantifier elimination (e.g. QEPCAD [Brown 2003], the constraint-based approach [Gulwani and Tiwari 2008]). We are currently examining a collection of

PCHAs with polynomial dynamics for which this direction is promising. Another direction of future research is related to the progress property. Although the basic principle is straightforward, the details of the progress proof in Sections 5.3 and 5.4 are quite involved. This is partly owing to the difficulty of finding the appropriate Lyapunov functions. In the future, we plan on investigating this further and use ideas from [Chandy et al. 2008] for the progress proof. A longer term goal is to integrate all these proof techniques within the TEMPO [TEM 2008] environment.

## 7. ACKNOWLEDGMENTS

The authors gratefully acknowledge Sumit Gulwani and Ashish Tiwari for letting us use their nonlinear solver for solving $\exists\forall$ problems.

## REFERENCES

2008. Tempo toolset, version 0.2.2 beta. `http://www.veromodo.com/tempo/`.

Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T. A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. 1995. The algorithmic analysis of hybrid systems. *Theoretical Computer Science 138,* 1, 3–34.

Bhatia, N. P. and Szegö, G. P. 1967. *Dynamical Systems: Stability Theory and Applications*. Lecture notes in mathematics, vol. 35. Springer-Verlag, Berlin; New York.

Brown, C. W. 2003. QEPCAD b: a program for computing with semi-algebraic sets using cads. *SIGSAM Bull. 37,* 4, 97–108.

Burdick, J. W., DuToit, N., Howard, A., Looman, C., Ma, J., Murray, R. M., and Wongpiromsarn, T. 2007. Sensing, navigation and reasoning technologies for the DARPA Urban Challenge. Tech. rep., DARPA Urban Challenge Final Report.

Chandy, K. M., Mitra, S., and Pilotto, C. 2008. Convergence verification: From shared memory to partially synchronous systems. In *Proceedings of Formal Modeling and Analysis of Timed Systems (FORMATS'08)*. LNCS, vol. 5215. Springer Verlag, 217–231.

DuToit, N. E., Wongpiromsarn, T., Burdick, J. W., and Murray, R. M. 2008. Situational reasoning for road driving in an urban environment. In *International Workshop on Intelligent Vehicle Control Systems (IVCS)*.

Fainekos, G. E., Girard, A., Kress-Gazit, H., and Pappas, G. J. 2009. Temporal logic motion planning for dynamic robots. *Automatica 45,* 2, 343–352.

Gulwani, S. and Tiwari, A. 2008. Constraint-based approach for analysis of hybrid systems. In *20th International Conference on Computer Aided Verification (CAV)*.

Henzinger, T. A., Kopke, P. W., Puri, A., and Varaiya, P. 1995. What's decidable about hybrid automata? In *ACM Symposium on Theory of Computing*. 373–382.

Kaynar, D. K., Lynch, N., Segala, R., and Vaandrager, F. 2005. *The Theory of Timed I/O Automata*. Synthesis Lectures on Computer Science. Morgan Claypool. Also available as Technical Report MIT-LCS-TR-917.

Kloetzer, M. and Belta, C. 2006. A fully automated framework for control of linear systems from ltl specifications. In *HSCC*. 333–347.

Lafferriere, G., Pappas, G. J., and Yovine, S. 1999. A new class of decidable hybrid systems. In *Hybrid Systems : Computation and Control*. Springer, 137–151.

Lynch, N., Segala, R., and Vaandrager, F. 2003. Hybrid I/O automata. *Information and Computation 185,* 1 (August), 105–157.

Mitra, S. 2007. A verification framework for hybrid systems. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA 02139.

Mitra, S., Wang, Y., Lynch, N., and Feron, E. 2003. Safety verification of model helicopter controller using hybrid Input/Output automata. In *Hybrid Systems: Computation and Control*, O. Maler and A. Pnueli, Eds. LNCS, vol. 2623. Springer, 343–358.

Owre, S., Rajan, S., Rushby, J., Shankar, N., and Srivas, M. 1996. PVS: Combining specification, proof checking, and model checking. In *Computer-Aided Verification, CAV '96*, R. Alur and T. A. Henzinger, Eds. Number 1102 in LNCS. Springer-Verlag, New Brunswick, NJ, 411–414.

Platzer, A. and Clarke, E. M. 2008. Computing differential invariants of hybrid systems as fixedpoints. In *CAV*, A. Gupta and S. Malik, Eds. Lecture Notes in Computer Science, vol. 5123. Springer, 176–189.

Prabhakar, P., Vladimerou, V., Viswanathan, M., and Dullerud, G. E. 2008. A decidable class of planar linear hybrid systems. In *Hybrid Systems: Computation and Control, 11th International Workshop, HSCC 2008, St. Louis, MO, USA, April 22-24, 2008. Proceedings.* LNCS, vol. 4981. Springer, 401–414.

Prajna, S. and Jadbabaie, A. 2004. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds. LNCS, vol. 2993. Springer, 477–492.

Prajna, S., Papachristodoulou, A., and Parrilo, P. A. 2002. Introducing SOSTOOLS: A general purpose sum of squares programming solver. In *Proceedings of the 41st IEEE Conf. on Decision and Control*. 741–746.

Sankaranarayanan, S., Sipma, H. B., and Manna, Z. 2008. Constructing invariants for hybrid systems. *Formal Methods in System Design 32,* 1, 25–55.

Vladimerou, V., Prabhakar, P., Viswanathan, M., and Dullerud, G. E. 2008. Stormed hybrid systems. In *ICALP (2)*. LNCS, vol. 5126. Springer, 136–147.

Wongpiromsarn, T., Mitra, S., Murray, R. M., and Lamperski, A. 2009. Periodically controlled hybrid systems: Verifying a controller for an autonomous vehicle. In *Hybrid Systems: Computation and Control*, R. Majumdar and P. Tabuada, Eds. LNCS, vol. 5469. Springer, 396–410.

Wongpiromsarn, T. and Murray, R. M. 2008. Distributed mission and contingency management for the DARPA urban challenge. In *International Workshop on Intelligent Vehicle Control Systems (IVCS)*.

## APPENDIX

## A. VEHICLE‖CONTROLLER AS A PCHA

Here we show that the composed automaton $\mathcal{A} = \mathsf{Vehicle}\|\mathsf{Controller}$ is a periodically controlled hybrid automaton. We define an automaton $\mathcal{A}'$ that is identical to $\mathcal{A}$ except that its variables, actions, and transition functions are renamed to match the definition of the generic PCHA of Figure 1.

*Variables.* $\mathcal{A}'$ has the following variables.

(a) A continuous state variable $s \triangleq \langle x, y, \theta, v, e_1, e_2, d\rangle$ of type $\mathcal{X} = \mathbb{R}^7$.

(b) A discrete state variable $loc \triangleq \langle brake, path, seg\rangle$ of type $\mathcal{L} = \mathtt{Tuple}[\{On, Off\}, \mathtt{Seq}[\mathbb{R}^2], \mathbb{N}]$.

(c) A control variable is $u = \langle a, \phi\rangle$ of type $\mathcal{U} = \mathbb{R}^2$.

(d) Two command variables $z_1 \triangleq brake$ of type $\mathcal{Z}_1 = \{On, Off\}$ and $z_2 = path$ of type $\mathcal{Z}_2 = \mathtt{Seq}[\mathbb{R}^2]$.

*Actions and transitions.* $\mathcal{A}$ has two input update actions, $\mathsf{brake}(b)$ and $\mathsf{plan}(p)$, and the command variables $z_1$ and $z_2$ store the values $b$ and $p$, respectively, when these actions occur.

An internal control action $\mathsf{main}$ occurs every $\Delta$ time, starting from time 0. That is, values of $\Delta_1$ and $\Delta_2$ as defined in a generic PCHA are $\Delta_1 = \Delta$ and $\Delta_2 = 0$. The control law function $g$ and the state transition function $h$ of $\mathcal{A}$ can be derived from the specification of $\mathsf{main}$ action in Figure 5. Let $g = \langle g_a, g_\phi\rangle$ where $g_a : \mathcal{L} \times \mathcal{X} \to \mathbb{R}$

and $g_\phi : \mathcal{L} \times \mathcal{X} \to \mathbb{R}$ represent the control law for $a$ and $\phi$, respectively, and are given by

$$g_a(l,s) = \begin{cases} a_{brake} & \text{if } l.brake = On \\ a_{max} & \text{if } l.brake = Off \ \wedge \ s_0.v < v_T \\ 0 & \text{otherwise} \end{cases}$$

$$g_\phi(l,s) = \frac{\phi_d}{|\phi_d|} \min(\delta \times s.v, |\phi_d|)$$

where $\phi_d = -k_1 s.e_1 - k_2 s.e_2$. Let $h = \langle h_{l,1}, h_{l,2}, h_{l,3}, h_{s,1}, \ldots, h_{s,7} \rangle$ where $h_{s,1}, \ldots, h_{s,7} : \mathcal{L} \times \mathcal{X} \times \mathcal{Z}_1 \times \mathcal{Z}_2 \to \mathbb{R}$ describe the discrete transition of $x$, $y$, $\theta$, $v$, $e_1$, $e_2$ and $d$ components of $s$, respectively, and $h_{l,1} : \mathcal{L} \times \mathcal{X} \times \mathcal{Z}_1 \times \mathcal{Z}_2 \to \{On, Off\}$, $h_{l,2} : \mathcal{L} \times \mathcal{X} \times \mathcal{Z}_1 \times \mathcal{Z}_2 \to \mathsf{Seq}[\mathbb{R}^2]$ and $h_{l,3} : \mathcal{L} \times \mathcal{X} \times \mathcal{Z}_1 \times \mathcal{Z}_2 \to \mathbb{N}$ describe the discrete transition of *brake*, *path* and *seg*, respectively. Then, the function $h$ is given by

$$\begin{aligned}
h_{s,1}(l,s,z_1,z_2) &= s.x, \ h_{s,2}(l,s,z_1,z_2) = s.y, \\
h_{s,3}(l,s,z_1,z_2) &= s.v, \ h_{s,4}(l,s,z_1,z_2) = s.\theta, \\
h_{s,5}(l,s,z_1,z_2) &= \begin{cases} s.e_1 & \text{if } l.path = z_2 \ \wedge \ s.d > 0 \\ \frac{1}{\|\vec{q}\|}\vec{q} \cdot \vec{r} & \text{otherwise} \end{cases}, \\
h_{s,6}(l,s,z_1,z_2) &= \begin{cases} s.e_2 & \text{if } l.path = z_2 \ \wedge \ s.d > 0 \\ s.\theta - \angle\vec{p} & \text{otherwise} \end{cases}, \\
h_{s,7}(l,s,z_1,z_2) &= \begin{cases} s.d & \text{if } l.path = z_2 \ \wedge \ s.d > 0 \\ \frac{1}{\|\vec{p}\|}\vec{p} \cdot \vec{r} & \text{otherwise} \end{cases}, \\
h_{l,1}(l,s,z_1,z_2) &= z_1, \ h_{l,2}(l,s,z_1,z_2) = z_2, \\
h_{l,3}(l,s,z_1,z_2) &= \begin{cases} 1 & \text{if } l.path \neq z_2 \\ l.seg + 1 & \text{if } l.path = z_2 \ \wedge \ s.d \leq 0 \\ l.seg & \text{otherwise} \end{cases}
\end{aligned}$$

where the temporary variable $\vec{p}$, $\vec{q}$ and $\vec{r}$ are computed as in the Controller specification based on the updated value of *path* and *seg*.

*Trajectories.* From the the state models of Vehicle and Controller automata specified on line 14 of Figure 4 and lines 48-50 of Figure 5, we see that $\mathcal{A}$ only has one state model. For any value of $l \in \mathcal{L}$, the continuous state $s$ evolves according to the differential equation $\dot{s} = f(s,u)$ where $f = \langle f_1, f_2, \ldots, f_7 \rangle$ and $f_1, \ldots, f_7 : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ are associated with the evolution of the $x$, $y$, $\theta$, $v$, $e_1$, $e_2$ and $d$ components of $s$, respectively. Using the definition of the control law function $g$

defined above, we can derive the following components of $f(s, g(l, s_0))$:

$$
\begin{aligned}
f_1(s, g(l, s_0)) &= s.v\cos(s.\theta),\ \ f_2(s, g(l, s_0)) = s.v\sin(s.\theta) \\
f_3(s, g(l, s_0)) &= f_6(s, g(l, s_0)) = \frac{s.v}{L}\tan(\frac{\phi_d}{|\phi_d|}\min(|\phi_d|, \delta s_0.v, \phi_{max})) \\
f_4(s, g(l, s_0)) &= \begin{cases} a_{brake} & \text{if } l.brake = On \ \wedge\ s.v > 0 \\ a_{max} & \text{if } l.brake = Off \ \wedge\ s_0.v < v_T \\ 0 & \text{otherwise} \end{cases} \\
f_5(s, g(l, s_0)) &= s.v\sin(s.e_2) \\
f_7(s, g(l, s_0)) &= -s.v\cos(s.e_2)
\end{aligned}
$$

where $\phi_d = -k_1 s_0.e_1 - k_2 s_0.e_2$.